Détection d'Anomalies Multiples par Apprentissage Automatique de Règles dans les Séries Temporelles

Inès Ben Kraiem

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse - Jean Jaurès*

Présentée et soutenue le *29/01/2021* **par :**

Ines BEN KRAIEM

Détection d'Anomalies Multiples par Apprentissage Automatique de Règles dans les Séries Temporelles

JURY

Anne LAURENT	Professeure, Université de Montpellier, France	Rapporteure
Pierre GANÇARSKI	Professeur, Université de Strasbourg, France	Rapporteur
Claudia RONCANCIO	Professeur, Université de Grenoble, France	Examinatrice
Karine ZEITOUNI	Professeure, Université de Versailles	Examinatrice
Faiza GHOZZI	Maître assistant, Université de Sfax	Examinatrice
Florence SEDES	Professeure, Université de Toulouse 3	Examinatrice
Olivier TESTE	Professeur, Université de Toulouse 2	Directeur de thèse
André PÉNINOU	Maître de conférence, Université de Toulouse 2	Co-directeur de thèse
François DOLVECK	Directeur adjoint du SGE, France	Invité

École doctorale et spécialité :

MITT : Domaine IT : Informatique et télécommunication

Unité de Recherche :

Institut de Recherche en Informatique de Toulouse (UMR 5505)

Directeur(s) de Thèse :

Olivier TESTE et *André PÉNINOU*

Rapporteurs :

Anne LAURENT et *Pierre GANCARSKI*

Ines Ben Kraiem

Détection d'Anomalies Multiples par Apprentissage Automatique de Règles dans les Séries Temporelles

Directeur de thèse :
Olivier Teste, professeur à l'université Toulouse 2 – Jean Jaurès

Résumé

Les outils de supervision et de monitoring sont communément utilisés dans l'industrie pour analyser les données issues de différents capteurs. Ces données sont souvent affectées par des événements inhabituels ou des changements temporaires et ont tendance à comporter des irrégularités et des valeurs aberrantes qui demandent des connaissances métiers du domaine et une intervention humaine pour être détectées. Dans de telles situations, la détection d'anomalies peut être un moyen crucial pour identifier les évènements anormaux et détecter les comportements inhabituels permettant ainsi aux experts d'agir rapidement et d'atténuer les effets d'une situation indésirable.

Dans cette thèse, nous avons focalisé sur l'utilisation de techniques d'apprentissage automatique dans le but d'automatiser et de consolider le processus de détection des anomalies dans les données de réseaux de capteurs. Ces données proviennent de capteurs se présentent sous forme de séries temporelles. Pour ce faire, nous avons défini deux objectifs principaux : la détection d'anomalies multiples et la génération de règles interprétables par l'être humain pour la détection d'anomalies.

Le premier objectif consiste à détecter différents types d'anomalies dans les données de capteurs. Dans les travaux de recherche existants, il existe un travail approfondi sur la détection d'anomalies. Cependant, la plupart des techniques recherchent des objets individuels qui sont différents des objets normaux ou bien des séquences de données, mais ne prennent pas en compte la détection de multiples anomalies. Pour résoudre cette problématique et atteindre notre premier enjeu, nous avons créé un système configurable de détection d'anomalies multiples qui est basé sur des motifs pour détecter les anomalies dans les séries temporelles. L'algorithme que nous proposons, Composition of Remarquable Point (CoRP), est basé sur le principe de recherche de motifs. Cet algorithme applique un ensemble de motifs afin d'annoter les points remarquables dans une série temporelle uni-variée, puis détecte les anomalies par composition de

motifs. Les motifs d'annotation et les compositions de motifs sont définis avec l'aide de l'expert du domaine. Notre méthode a l'avantage de localiser et de catégoriser les différents types d'anomalies détectées.

Le deuxième objectif de la thèse est la génération de règles interprétables et intelligibles par les experts pour la détection d'anomalies. Pour ceci, nous avons proposé un algorithme, Composition based Decision Tree (CDT), qui permet de produire automatiquement des règles ajustables et modifiables par les experts. Pour ce faire, nous avons conçu une modélisation variable des motifs de détection des points remarquables pour labéliser les séries temporelles. Sur la base de la série temporelle étiquetée, un arbre de décision est construit en considérant les nœuds comme des compositions de motifs. Enfin, l'arbre est converti en un ensemble de règles de décision, compréhensibles par les experts. Nous avons aussi défini une mesure de qualité pour les règles produites.

Nous avons testé les performances de CoRP et CDT avec des compétiteurs, sur des données réelles et des données issues de la littérature (benchmarks). Les deux méthodes font preuve d'efficacité pour la détection d'anomalies multiples. Les résultats ont une bonne précision offrant un taux élevé de détection avec un faible taux de faux positifs.

Les travaux développés dans cette thèse ont été menés dans le cadre du projet neoCampus et financés par le Service de Gestion et d'Exploitation rattaché au rectorat de Toulouse.

Institut de Recherche en Informatique de Toulouse – UMR 5505 CNRS

Université Toulouse 3 – Paul Sabatier, 118 route de Narbonne, F-31062 Toulouse cedex 9

Ines Ben Kraiem

Detection of Multiple Anomalies by The Automatic Learning of Rules in Time Series

Supervisor:
Olivier Teste, Professor at Toulouse 2 University – Jean Jaures

Abstract

Supervision and monitoring tools are commonly used in the industry to analyze data from different sensors. These data are often affected by unusual events or temporary changes and tend to contain irregularities and outliers that require business knowledge and human intervention to be detected. In such situations, anomaly detection can be a crucial way to identify abnormal events and detect unusual behavior, allowing experts to act quickly and mitigate the effects of an undesirable situation.

In this thesis, we focused on the use of automatic learning techniques in order to automate and consolidate the process of detecting anomalies in sensor network data. These data come from sensors and are presented in the form of time series. To do this, we have defined two main objectives: the detection of multiple anomalies and the generation of interpretable rules by humans for the detection of anomalies.

The first objective is to detect different types of anomalies in the sensor data. In the existing research, there is extensive work on anomaly detection. However, most techniques look for individual objects that are different from normal objects or a sequence of data but do not take into consideration the detection of multiple anomalies. To solve this problem and reach our first issue, we have created a configurable multiple anomaly detection system that is based on patterns to detect anomalies in time series. The algorithm we propose, Composition of Remarquable Point (CoRP), is based on the principle of pattern search. This algorithm applies a set of patterns to annotate remarkable points in a uni-varied time series, then detects anomalies by pattern composition. Annotation patterns and pattern compositions are defined with the help of the subject matter expert. Our method has the advantage of locating and categorizing the different types of anomalies detected.

The second objective of the thesis is the generation of rules that can be interpreted

and understood by experts for the detection of anomalies. For this, we have proposed an algorithm, Composition based Decision Tree (CDT), which automatically produces rules that can be adjusted and modified by experts. To do this, we have designed variable modeling of the detection patterns of remarkable points to label the time series. Based on the labeled time series, a decision tree is constructed by considering the nodes as compositions of patterns. Finally, the tree is converted into a set of decision rules, understandable by experts. We have also defined a quality measure for the rules produced.

We tested the performance of CoRP and CDT with competitors, on real data and data from the literature (benchmarks). Both methods are effective in detecting multiple anomalies. The results have good precision offering a high detection rate with a low false-positive rate.

This PhD was supported by the Management and Exploitation Service (SGE) of the Rangueil campus attached to the Rectorate of Toulouse and the researchis made in the context of the neOCampus project (Paul Sabatier University,Toulouse).

Institut de Recherche en Informatique de Toulouse – UMR 5505 CNRS

Université Toulouse 3 – Paul Sabatier, 118 route de Narbonne, F-31062 Toulouse cedex 9

Remerciements

À l'issue de ce travail de doctorat, je désire témoigner ma reconnaissance à un grand nombre de personnes avec qui j'ai travaillé tout au long de mon parcours. Leur générosité et leur intérêt manifestés à l'égard de ma recherche m'ont permis de progresser dans mon parcours d'apprenti chercheur et je souhaite ici leur affirmer à quel point ils m'ont apporté.

Je tiens tout d'abord à remercier les membres extérieurs du jury. Ainsi, ma reconnaissance va à Pr. Anne LAURENT et à Pr. Pierre GANÇARSKI qui m'ont fait l'honneur d'être rapporteurs de ce mémoire. Je remercie également Pr. Florence SEDES, Pr. Karine ZEITOUNI et Pr. Claudia RONCANCIO, d'en être les examinatrices. Je les remercie pour leur évaluation scientifique et leur travail de synthèse. Qu'ils soient assurés de mon très grand respect.

Je tiens à exprimer ma sincère gratitude à mon directeur de thèse, Pr. Olivier TESTE pour le soutien continu de mes études de doctorat et des recherches connexes, pour sa patience, sa motivation et ses immenses connaissances. Il m'a fait bénéficier de son recul sur de nombreux domaines de l'informatique fondamentale comme appliquée. Je tiens ici à lui témoigner mon admiration, ma gratitude et mon profond respect.

Je dois beaucoup à Dr. André PENINOU, co-encadrant de cette thèse, qui a été pour moi mon parent académique. Passionné par ses activités d'enseignant comme de chercheur, il a contribué grandement à la réalisation de ces travaux. Je n'aurais pas pu imaginer avoir un meilleur encadrant pour mon doctorat. Qu'il soit assuré de ma reconnaissance pour son soutien et ses nombreux encouragements, ainsi que du plaisir que j'ai à travailler avec lui.

Je suis reconnaissante envers Dr. Faiza GHOZZI, co-encadrante de cette thèse, à plusieurs égards. C'est grâce à elle que j'ai eu cette opportunité de travailler et de rencontrer mes encadrants et de faire cette thèse en France. Alors que j'étais étudiante en cycle ingénieur en Tunisie, elle m'a offert cette opportunité de découvrir le domaine de la recherche en dehors de mon pays natal. C'est également grâce à Madame GHOZZI que j'ai découvert l'IRIT, où j'ai effectué mon premier stage de cycle ingénieur. J'espère

être à la hauteur de sa confiance et qu'elle trouve dans ce travail l'expression de ma profonde gratitude.

Je souhaite remercier Geoffrey ROMAN-GIMNEZ, chercheur postdoctoral à l'IRIT, pour sa contribution à mes recherches. Notre collaboration a été propice à une réflexion approfondie sur notre domaine de recherche. Je tiens à lui exprimer mes remerciements pour ses efforts et pour ces qualité humaines, pour cette expérience enrichissante en participant à ce travail.

Ma gratitude va également à Monsieur Hervé CROS, ancien responsable du service électromécanique, qui m'a encadré au SGE. Je tiens à le remercier pour sa disponibilité et ses conseils avisés. Je souhaite également remercier Monsieur François DOLVECK, directeur adjoint du SGE, pour l'intérêt et le regard critique qu'il a toujours porté à mes recherches et les moyens qu'il a mis en œuvre pour me donner accès aux données. Merci de m'avoir fait confiance. Un grand merci à tous les personnels du SGE pour votre accueil pendant ces trois ans ! À ceux qui m'ont particulièrement supporté quand j'avais des soucis et à tous ceux avec qui j'ai partagé de très bons moments.

Au fil de ces trois années passées à l'IRIT, j'ai eu la chance d'être entourée de personnes généreuses et passionnées. Je souhaite tout particulièrement remercier Wafa ABDELGHANI Oihana COUSTI pour les bons moments que nous passons au quotidien. Nos discussions du déjeuner, nos pauses cafés, souvent désinvoltes mais parfois très sérieuses, sont autant d'escapades dont j'ai besoin. L'ambiance au laboratoire ne serait pas la même sans vous.

Je souhaite également exprimer à mes amis à quel point leur présence m'est indispensable. Je remercie Carmela et Ahmed qui ont su me comprendre et m'épauler dans les moments difficiles, faire preuve de patience et d'un amour sans réserve.

Enfin, un immense merci à ma famille pour leur soutien sans faille tout au long de mes études. En particulier à mon père, ma mère, ma soeur et ma nièce pour leur amour sans condition et leur confiance en moi et pour avoir accepté le fait que je sois à l'étranger depuis trois ans. On ne choisit certes pas sa famille, mais je ne vous changerai pour rien au monde.

Ces travaux de thèse ont été financés par le Service de Gestion et d'Exploitation (SGE) du campus de Rangueil rattaché au Rectorat de Toulouse et la recherche est menée dans le cadre du projet neOCampus (Université Paul Sabatier, Toulouse).

Table des matières

Introduction générale

> "Artificial intelligence is defined as the branch of science and technology that is concerned with the study of software and hardware to provide machines the ability to learn insights from data and the environment, and the ability to adapt in changing situations with high precision, accuracy and speed."
>
> *Alison Ray (2018)*

Contexte de travail

La supervision (« monitoring ») des réseaux de capteurs est une activité importante dans l'industrie (Cateni *et al.*, 2008; Xu et Balazinska, 2011). Ces réseaux de capteurs produisent des ensembles de données, le plus souvent estampillées temporellement. Dans le domaine de l'habitat, les experts (les ingénieurs ou les techniciens de maintenance) explorent les données issues de différents capteurs afin de faire le suivi des consommations énergétiques dans les bâtiments, détecter les défaillances et les défauts de compteurs, et surveiller les alarmes. Comme dans tout système de supervision, l'expertise et l'intervention des experts sont primordiales dans la détection *d'anomalies* afin d'aider à la prise de décisions et choisir les actions à mener.

Afin d'identifier les problèmes, les experts analysent manuellement les différentes courbes formées à partir des *séries temporelles* que forment les données et distinguent les profils normaux des profils anormaux. Ils détectent ainsi les points remarquables, qui sont hors de la plage normale, et, par analyse des données autour de ces points,

ils sont capables de donner une identification du phénomène et une localisation précise (points ou séquences de points) des anomalies. Ce processus d'investigation et de traitement de données est une tâche complexe qui nécessite une bonne connaissance du domaine. De plus l'investigation menée souvent manuellement et l'analyse de données pour la décision prennent énormément de temps. Les raisons sous-jacentes sont liées à la variété des types d'anomalies pouvant être observées dans les déploiements réels et le volume croissant des données qui rendent l'analyse des experts complexe (Sharma *et al.*, 2010; Kiani *et al.*, 2020). Compte tenu de ces observations, une direction de recherche importante est la détection automatisée des anomalies, la recherche de motifs et l'extraction des règles afin d'aider les experts à prendre les décisions adéquates.

Dans nos travaux, nous nous intéressons à cette problématique de la détection automatique d'anomalies dans les séries temporelles, qui apparaît comme étant le moyen pour identifier les évènements anormaux et détecter des comportements qui ne sont pas conformes au comportement attendu et de reconnaître quelles valeurs sont problématiques parmi toutes les données. Au-delà de la supervision des réseaux de capteurs, la détection d'anomalies présente un enjeu majeur dans de multiples applications telles que la détection de fraude par carte de crédit, la supervision de l'état de santé (comme les moniteurs de fréquence cardiaque), la détection des intrusions sur les réseaux, les applications financières et le marketing, la surveillance de l'habitat, l'analyse du trafic web et bien d'autres (Cateni *et al.*, 2008; Chandola *et al.*, 2009; Gupta *et al.*, 2013; Aggarwal, 2015). Par exemple, dans le domaine de la santé, une situation médicale anormale au niveau du cœur d'un patient peut être détectée en identifiant des anomalies dans la série chronologique correspondant aux enregistrements d'électrocardiogramme (ECG) du patient. Un autre exemple convienne les applications de surveillance d'habitats avec des réseaux de capteurs, où une augmentation inattendue de la consommation d'énergie dans un bâtiment génère à un pic dans les données qui peut être considérée comme une anomalie à détecter. Dans un autre contexte, un fournisseur de carte de crédit essaye d'identifier les transactions frauduleuses. Si le système enregistre un achat de plusieurs milliers d'euros alors que le client a l'habitude d'utiliser sa carte pour des petits achats, il y a de fortes suspicions pour qu'il se soit fait voler sa carte ou ses identifiants de paiement.

Le problème de la détection d'anomalies dépend de plusieurs facteurs tels que la nature des données, la disponibilité des données étiquetées (labels), les types d'anomalies à considérer et le résultat à fournir pour le système en question. Ces facteurs déterminent le choix de l'algorithme à utiliser. Souvent, ces facteurs sont déterminés par le domaine d'application dans lequel les anomalies doivent être détectées. La figure .1 montre les composants clés mentionnés ci-dessus associés à la problématique de la

détection d'anomalies. Nous allons aborder en détails ces notions dans la partie I.

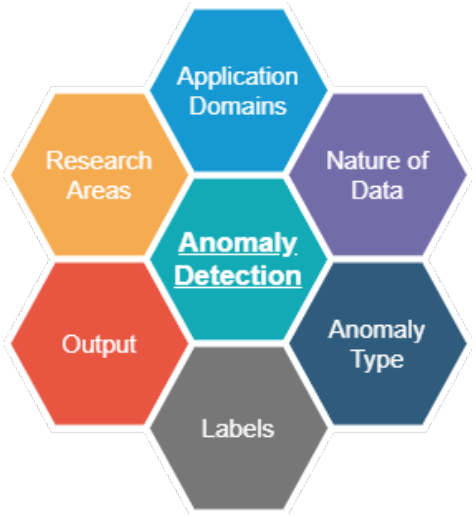


Figure .1 – Les caractéristiques d'un problème de détection d'anomalies.

Dans cette thèse, nous identifions différents types d'anomalies observées dans des déploiements réels. Nous étudions les techniques de détection d'anomalies et les types d'apprentissage et nous nous concentrons sur une formulation de problème de deux manière : la détection d'anomalies non-supervisée et supervisée (Xu et Balazinska, 2011). Nous proposons une nouvelle technique de détection d'anomalies par apprentissage automatique de motifs (« patterns » en anglais) prédéfinis sur les données de nature temporelle, en particulier pour les séries temporelles uni-variées.

Domaine d'application : Réseau de capteurs du SGE

Cette thèse a été financé par le Service de Gestion et d'Exploitation (SGE) du campus de Rangueil rattaché au Rectorat de Toulouse et la recherche est menée dans le cadre du projet neOCampus (Université Paul Sabatier, Toulouse). Le but est d'améliorer de façon significative le processus de prise de décision et l'exploitation des données de compteurs/capteurs qui représentent une source d'informations essentielle au SGE. Ces données sont constituées de séries temporelles ; i.e. des remontées de mesures de capteurs/compteurs collectées séquentiellement dans le temps à intervalles réguliers.

Le SGE est un service du rectorat spécialisé dans la gestion, l'exploitation et l'investissement des réseaux mutualisés de Rangueil. Il gère les données liées aux différentes installations en termes de fluides (énergie, eau, air comprimé) sur différents campus. Plus de 1000 compteurs sont répartis dans 255 bâtiments.

Les données proviennent de capteurs (e.g., température, vannes, pression) ou de

compteurs (e.g., eau chaude, électricité). Les différents relevés sont conservés automatiquement à intervalles réguliers (e.g., toutes les 30', 15 minutes, 1 heure). Par conséquent, chaque relevé concerne une valeur avec un horodatage. Le SGE dispose d'outils de supervision temps-réel permettant à l'opérateur de visualiser et de piloter la production, la gestion d'alarmes, l'affichage de tendances, l'archivage de données et l'acquisition de données.

L'objet de cette thèse est de proposer des méthodes automatisées de détection d'anomalies appliquées aux données de compteurs, des séries temporelles régulières, et d'en tirer des règles de détection afin d'obtenir des analyses plus juste.

Objectifs de la thèse

Notre objectif est d'étudier et de proposer des techniques pour la détection d'anomalies à partir des séries temporelles produites par les réseaux de capteurs. Les techniques proposées sont basées sur des concepts d'analyse de données et d'apprentissage automatique (Chandola *et al.*, 2009; Gupta *et al.*, 2013; Aggarwal, 2015). Comme indiqué précédemment, nous nous concentrons ici sur les tâches de détection et d'extraction de règles. Cette thèse a donc deux enjeux majeurs :

— Détection d'anomalies : Le premier enjeu réside dans la détection de multiple anomalies de types différents. La détection devrait être automatique et non supervisée. De plus, la solution proposée doit être efficace c'est à dire avec un faible taux d'erreurs et en nécessitant peu de ressources de calcul. Le but principal est de localiser automatiquement les anomalies détectées tout en expliquant les catégories ou les types d'anomalies trouvés.

— Extraction de règles : Le deuxième enjeu consiste à produire des règles de détection compréhensibles et intelligibles par les experts afin de les aider à mieux analyser et traiter les situations anormales. La solution doit être robuste en garantissant une détection d'anomalies élevées avec un faible taux de mauvaises classifications et des règles interprétables par les experts.

Contributions de la thèse

Pour répondre aux objectifs évoqués ci-dessus, nous nous intéressons dans cette thèse à deux facettes de la détection d'anomalies dans les séries temporelles, en mode non-supervisé et en mode supervisé :

— Nous proposons une première contribution qui est une approche configurable non supervisée basée sur la modélisation de *motifs* de détection d'anomalies multiples dans des séries temporelles uni-variées. Notre algorithme intitulé CoRP (Composition of Remarkable Points), applique un ensemble de motifs afin d'annoter les points remarquables dans une série temporelle uni-variée. Les points remarquables sont les points qui ont un comportement inhabituel en comparant avec le reste des données. Ces points sont détectés d'habitude par les experts manuellement en regardant les écarts entre les points successifs. Ensuite, CoRP détecte les anomalies par composition de motifs à travers une grammaire que nous avons proposée. La sortie de CoRP est la localisation des anomalies avec les types adéquats. Les motifs d'annotation et les compositions de motifs sont définis avec l'aide de l'expert du domaine. Cette approche permet de détecter finement les anomalies à savoir des points spécifiques anormaux dans les séries temporelles et de les catégoriser.

— La deuxième contribution est basée sur une méthode d'apprentissage automatique supervisée. Premièrement, nous proposons une modélisation variable des motifs de détection des points remarquables basées sur une formalisation de neuf variations anormales correspondant aux types d'anomalies observées dans les déploiements réels. Deuxièmement, nous proposons une version adaptée des arbres de décision, intitulé les Composition-based Decision Tree (CDT), pour produire des règles interprétables par l'homme. Compte tenu de l'étiquetage de séries temporelles, CDT construit un arbre de décision, en considérant les nœuds comme des compositions de motifs avec le gain d'information le plus élevé. L'entrée du CDT est construite en créant des fenêtres glissantes de taille fixe, où les hyperparamètres sont automatiquement calculés via une optimisation bayésienne. L'arbre est ensuite converti en un ensemble de règles de décision, pour lesquelles une mesure de qualité est définie. Cette mesure vise à garantir l'interprétabilité de la composition en tenant compte de la longueur de la règle et de son nombre d'étiquettes. Notre approche permet de générer automatiquement des règles intelligibles et compréhensibles par les experts. Nous appliquons également des simplifications booléennes afin de simplifier les règles de détection avant de les présenter aux experts.

Toutes nos expérimentations ont été menées d'une part sur des données réelles issues des réseaux de capteurs du SGE et d'autre part, sur des données d'autres domaines d'application issus de la littérature scientifique. Nous montrons que notre première méthode est précise pour classifier les anomalies par rapport aux autres méthodes de la littérature. Nous montrons que notre deuxième méthode est robuste par rapport aux algorithmes existants notamment en cas d'anomalies multiples, ce qui est le cas dans les applications réelles. Un autre atout de notre approche réside dans le plus faible

nombre de règles interprétables générées (ce qui simplifie l'effort d'interprétation des experts).

Organisation du mémoire

Le reste de ce manuscrit est organisé comme suit.

La partie 1 est dédiée à l'étude des travaux de la littérature liés aux travaux de cette thèse. Elle comporte 2 chapitres. Dans le premier chapitre, nous présentons une description des concepts fondamentaux nécessaires à la compréhension de notre travail. Dans le deuxième chapitre, nous décrivons les domaines d'application et les techniques de détection d'anomalies, avec une attention particulière sur la problématique de la détection d'anomalies dans les séries temporelles. Nous détaillons les différents types d'anomalies traités dans les travaux de la littérature et nous les comparons avec les anomalies observées dans les déploiements réels. Nous présentons les différentes stratégies de détections d'anomalies et les méthodes d'évaluation. Enfin, nous comparons notre contribution par rapport aux algorithmes de l'état de l'art sur la base d'un ensemble de critères qui ont été relevés dans ce chapitre.

La partie 2 est consacrée à nos contributions. Notre approche est basée sur les motifs pour la détection d'anomalies. Elle comporte 3 chapitres. Le premier chapitre donne un aperçu du contexte de la recherche et de nos motivations, puis aborde quelques notions utiles pour la suite des chapitres et rappelle les types d'anomalies que nous cherchons à détecter. Le deuxième chapitre présente notre première approche « CoRP : Composition of Remarquable Point » pour la détection multiple d'anomalies en utilisant des motifs. Le troisième chapitre présente notre deuxième approche « CDT : Composition-based Decision Tree » pour la génération automatique de règles intelligibles pour la détection d'anomalies.

La partie 3 est dédiée aux expérimentations menées pour valider les deux contributions proposées. Elle comporte 3 chapitres. Nous présentons dans le premier chapitre les données du Service de Gestion et d'Exploitation puis les données issues de la littérature sur lesquelles nous avons réalisé nos expérimentations. Les chapitres 2 et 3 présentent les résultats d'expérimentations de CoRP et CDT respectivement sur les différents ensembles de données.

La dernière partie conclut ce document et dresse de futures perspectives.

Publications liées à la thèse

Voici une liste des publications publiées au cours de cette thèse.

- Conférences internationales
  - Ines Ben Kraiem, Faiza Ghozzi, André Péninou, Geoffrey Roman-Jimenez and Olivier Teste : Human-Interpretable Rules for Anomaly Detection in Time-series. In Proceedings of the 24th International Conference on Extending Database Technology, EDBT 2021.
    Cet article correspond aux contributions présentées dans le chapitre II.3. Il présente notre deuxième contribution sur la détection d'anomalies, CDT. Il présente l'extension de CDT en intégrant la simplification des règles et l'évaluation de qualité des règles. Il intègre l'optimisation bayésienne pour automatiser les hyper-paramètres de CDT.
  - Ines Ben Kraiem, Faiza Ghozzi, André Péninou, Geoffrey Roman-Jimenez and Olivier Teste : Automatic Classification Rules for Anomaly Detection in Time-series. In Proceedings of the 2020 14th International Conference on Research Challenges in Information Science (RCIS) (2020).
    Cet article correspond aux contributions présentées dans le chapitre II.3. Il présente notre deuxième contribution sur la détection d'anomalies, CDT. Il présente CDT et son évaluation sur différents ensembles de données en comparant avec des techniques connues dans l'apprentissage automatique.
  - Ines Ben Kraiem, Faiza Ghozzi, André Péninou, Olivier Teste : Pattern-based Method for Anomaly Detection in Sensor Networks. ICEIS 2019 : 104-113 **("best student paper award")**.
    Cet article correspond aux contributions présentées dans le chapitre II.2. Il présente notre première contribution sur la détection d'anomalies, CoRP.
- Chapitre du livre
  - Ines Ben Kraiem, Faiza Ghozzi, André Péninou, Olivier Teste. (2020) CoRP : A Pattern-Based Anomaly Detection in Time-Series. In : Enterprise Information Systems. ICEIS 2019 (extension). Lecture Notes in Business Information Processing, vol 378. Springer, Cham. https ://doi.org/10.1007/978-3-030-40783-4_20
    Cet article correspond aux contributions présentées dans le chapitre II.2. Il présente notre première contribution sur la détection d'anomalies, CoRP. Il présente une extension de l'article publié à ICEIS avec en particulier une évaluation expérimentale plus approfondie.

- Conférences nationales françaises

  — Ines Ben Kraiem, Faiza Ghozzi, André Péninou, Olivier Teste : Méthode basée sur les patterns pour la détection simultanée d'anomalies multiples dans les réseaux de capteurs. INFORSID 2019 : 239-254.
  Cet article correspond aux contributions présentées dans le chapitre II.2. Il présente notre première contribution sur la détection d'anomalies, CoRP.

Première partie

État de l'art

1 Exploration des séries temporelles

« Comme l'apprend vite tout bon rédacteur, c'est justement ce qui est évident qui doit être souligné — sinon on passera à côté. »

Peter Ferdinand Drucker (1909 — 2005)

Table des matières

Cette première partie du mémoire présente la problématique de la détection d'anomalies dans les séries temporelles et les travaux les plus pertinents de la littérature permettant de la résoudre. Nous présentons tout d'abord, les concepts fondamentaux de notre travail. Puis, nous décrivons les domaines d'application et les techniques de détection d'anomalies. Nous détaillons les différents types d'anomalies traités dans les travaux de la littérature et nous les comparons avec les anomalies observées dans les déploiements réels. Nous présentons les différentes stratégies de détections d'anomalies et les méthodes d'évaluation. Enfin, nous comparons notre contribution par rapport aux algorithmes de l'état de l'art sur la base d'un ensemble de critères qui ont traversé ce chapitre.

Dans de très nombreux domaines scientifiques, les mesures sont effectuées au fil du temps. Ces observations conduisent à une collection de données organisées sous forme de séries chronologiques ou séries temporelles (« time-series » en anglais). Le but de l'exploration de données de séries chronologiques est d'essayer d'extraire toutes les connaissances significatives dans ces données. Même si les humains ont une capacité naturelle à effectuer ces tâches, cela reste un problème complexe que les ordinateurs peuvent effectuer rapidement sur de vastes quantités de données. L'apprentissage automatique qui a connu un important essor durant cette décennie est une voie d'amélioration que nous adoptons dans nos travaux. Ce chapitre présente le contexte scientifique de nos travaux. Il décrit tout d'abord les données traitées qui sont les séries chronologiques. Il présente ensuite l'exploration des modèles dans ces données et son utilisation avec l'apprentissage automatique.

1.1 Série temporelle

Une série temporelle est un ensemble de variables à valeurs réelles collectées (suite d'observations) séquentiellement dans le temps à un intervalle régulier ou irrégulier. Ces observations représentent des mesures associées à un horodatage indiquant l'horaire de sa collecte (Brockwell *et al.*, 2002).

Les exemples de séries chronologiques sont des données météorologiques telles que la température ou les précipitations ; des données économiques comme les cours des actions et des données médicales. Une série chronologique permet d'analyser l'effet d'événements cycliques, saisonniers et irréguliers sur l'élément de données mesurées (Archana et Pawar, 2014). Selon le nombre de variables et la dépendance entre elles, une série temporelle peut être uni-variée ou multivariée.

— Une série temporelle uni-variée est une séquence de mesures d'une seule variable

qui dépend du temps.

— Une série temporelle multivariée est une séquence de mesures de plusieurs variables. Les variables sont co-dépendantes et dépendent également du temps (Cheng *et al.*, 2009). Dans ce cas les observations sont vectorielles.

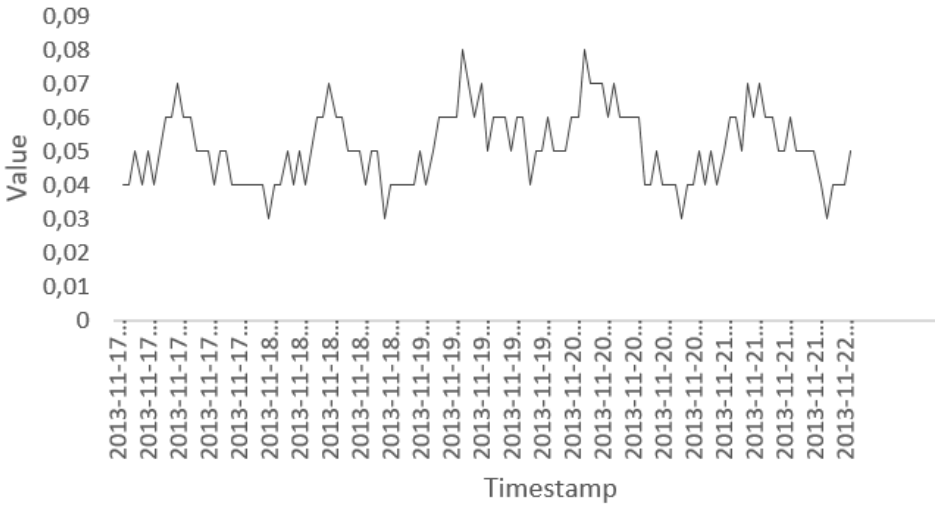


Figure I.1.1 – Exemple de série temporelle uni-variée correspondante à la consommation énergétique d'un bâtiment calculé à travers les relevés d'index d'un compteur.

La figure I.1.1 est un exemple d'une série temporelle uni-variée. Elle montre l'évolution de la consommation énergétique d'un bâtiment au cours du temps. Ces valeurs, numériques, sont calculées à partir des relevés d'un compteur ou index que le SGE gère.

Dans notre travail, nous traitons ce type de séries temporelles uni-variées issues de différents compteurs. Nous réalisons nos expérimentations notamment sur les données d'index ou de consommation. Ces mesures sont régulières dans le temps. Cependant, elles peuvent contenir des données manquantes à cause d'une panne dans les réseaux de capteurs ce qui engendrent par la suite différentes anomalies. Pour les experts, il est important de détecter les valeurs incohérentes et les anomalies au lieu de les supprimer pour faire leurs analyses. Il convient alors de traiter les séries temporelles sans faire une imputation des données manquantes.

1.1.1 Extraction de motifs

L'exploration de modèles ou motifs (« patterns ») fait référence à une méthode d'exploration de données qui consiste à trouver des modèles existant dans les données.

L'exploration de modèles fréquents ou peu fréquents, l'exploitation séquentielle de modèles, l'extraction d'ensemble d'objets (itemset) relèvent également de l'exploration de modèles. Les motifs fréquents sont des sous-séries qui apparaissent un nombre significatif de fois. Pour rechercher de tels motifs, il faut que son support soit supérieur ou égal à un seuil minimal défini par l'utilisateur. Ce principe est très utilisé dans les règles d'association comme par exemple l'algorithme très connu Apriori créer par Agrawal et Srikant (1994), pour rechercher les motifs les plus fréquents.

La détection des modèles aberrants, peu fréquents, peut-être plus importante dans de nombreuses séquences que celle des modèles d'analyse réguliers et plus fréquents. Un changement de comportement du client, un rythme cardiaque ECG inhabituel, des modèles surprenants dans les séquences de protéines, etc., représentent des modèles aberrants pouvant indiquer des anomalies.

Les séries chronologiques peuvent contenir des motifs spécifiques qui seraient pertinents pour l'analyse de données. L'idée de l'extraction de formes dans les séries chronologiques n'est pas nouvelle et a été abordée de différentes manières. La série chronologique est divisée en segments (Keogh *et al.*, 2004) et les segments sont classés en classes de modèles (Horst et Abraham, 2004). La classification peut être basée sur n'importe quel algorithme d'apprentissage automatique (ML) tel que K-means, Support Vector Machine (SVM), arbres de décision ou forêts aléatoires, réseaux de neurones etc. (Zhou *et al.*, 2015). Les séries temporelles saisonnières également, en raison de leur périodicité, sont de bons candidats pour être analysées via les outils d'extraction de motifs.

Dans notre domaine de recherche, nous cherchons à détecter la rareté. Par conséquent, nous nous focalisons sur la détection de motifs peu fréquents constituant des informations importantes pour la détection d'anomalies dans les séries temporelles.

1.1.2 Les fenêtres glissantes

L'algorithme de fenêtre glissante est une méthode de segmentation de données de séries chronologiques bien connue (Hota *et al.*, 2017). La fenêtre glissante est une approximation temporaire de la valeur réelle des données de la série chronologique (Hota *et al.*, 2017).

La figure I.1.2 illustre un exemple du processus de fenêtre glissante avec une taille de fenêtre égale à 4. Chaque x représente l'observation quotidienne des données de séries chronologiques (1,2,3...N).
La fenêtre couvrait de 1 à 4 (rectangle noir) représente les données historiques de 4 jours sont ainsi utilisées pour prédire la valeur du jour suivant. La fenêtre (rectangle

rouge) glisse de gauche à droite d'un jour, un pas de 1, pour couvrir encore 4 jours (de 2 à 5) pour prédire le jour suivant. Le processus est répété jusqu'à ce que toutes les données de la série chronologique soient ainsi segmentées.

Nous allons utiliser ce principe comme pré-traitement des séries temporelles pour l'extraction de règles à partir des segments (fenêtres).

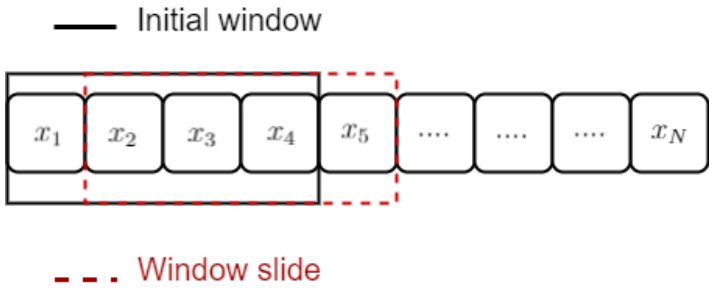


Figure I.1.2 – Exemple de processus du principe de sliding window.

1.2 Apprentissage automatique

L'apprentissage automatique (ML « Machine Learning » en anglais) (Witten *et al.*, 2016) est un domaine d'analyse de données, dans lequel l'algorithme apprend à partir des données. Le ML vise à imiter l'apprentissage humain, le raisonnement et la prise de décision. Contrairement aux systèmes experts qui sont des systèmes automatisés basés sur l'expertise humaine, les systèmes basés sur le ML s'appuient entièrement sur le système pour apprendre des données. "L'intelligence" des algorithmes ML vient du traitement itératif des données jusqu'à la convergence d'une fonction objectif. Le ML permet de construire des systèmes autonomes constituant une intelligence artificielle. Il y a principalement deux phases dans un algorithme basé sur l'apprentissage automatique : une phase d'apprentissage où le système apprend à partir des données et une phase de test lorsque l'algorithme applique les connaissances acquises à de nouveaux échantillons. Le système peut également continuer d'apprendre des nouvelles données pour augmenter ses performances. Dans la suite, nous nous intéressons aux différentes approches basées sur l'apprentissage automatique pour la détection d'anomalies.

Ces concepts de base constituent le périmètre de notre domaine de recherche relatif à la détection automatique d'anomalies dans les séries temporelles. Nous dressons dans la section suivante un état de l'art relatif aux techniques de détection d'anomalies qui sont basées sur l'apprentissage automatique.

2 Détection d'anomalies dans les séries temporelles

« Il n'y a pas une méthode unique pour étudier les choses. »

Aristote (384 —322)

Table des matières

2.1 Introduction

Dans ce chapitre, nous introduisons la problématique de la détection d'anomalies dans les séries temporelles. Nous présentons les types d'anomalies traitées dans la littérature et nous nous focalisons sur les types d'anomalies rencontrées dans des déploiements réels de capteurs. Ensuite, nous présentons une étude synthétique des domaines d'application et des techniques de détection d'anomalies. Nous détaillons plusieurs travaux illustrant ces techniques avec les principaux concepts associés. Enfin, nous concluons ce chapitre par les méthodes et mesures d'évaluation de ces techniques dans un contexte expérimental.

2.2 Contexte

Les progrès technologiques récents nous permettent de collecter une grande quantité de données au fil du temps dans divers domaines d'application. Les observations enregistrées de manière ordonnée dans le temps constituent une série chronologique/temporelle (« time-series »). L'exploration de données de séries temporelles vise à extraire toutes les connaissances significatives de ces données, les comportements normaux ou anormaux dans le temps constituant des indications sur les conditions de fonctionnement anormales du système monitoré (Gupta *et al.*, 2013). Par exemple un défaut de rotation du moteur d'avion, un défaut sur une ligne de production ou une hausse de température sont des indications pouvant signifier un dysfonctionnement.

La détection d'anomalies apparaît comme étant le moyen pour identifier les évènements anormaux et trouver des modèles dans les données qui ne correspondent pas au comportement normal du système (Chandola *et al.*, 2009). L'une des premières études sur ce sujet, a été menée par Fox (1972). Deux types de valeurs aberrantes dans les séries temporelles univariées ont été définis : le type I, qui affecte une seule observation et le type II, qui affecte à la fois une observation particulière et les observations ultérieures. Ce travail a d'abord été étendu à quatre types aberrants (Tsay , 1988), puis au cas des séries temporelles multivariées (Tsay *et al.*, 2000).

Depuis, de nombreuses définitions du terme aberrant et de nombreuses méthodes de détection ont été proposées dans la littérature. Cependant, à ce jour, il n'y a toujours pas de consensus sur les termes utilisés (Carreno *et al.*, 2019; Carrera *et al.*, 2019) ; selon les applications et les domaines de recherche les observations aberrantes sont dénommées de différentes manières, tels qu'anomalies, observations discordantes, discordes, exceptions, aberrations, surprises, particularités ou contaminants (Blázquez-

García *et al.*, 2020). Parmi ceux-ci, les termes les plus courants dans la littérature sont anomalies et valeurs aberrantes. Selon Chandola *et al.* (2009), les anomalies font référence à « des modèles dans les données qui ne sont pas conformes à une notion bien définie du comportement normal ». Une autre analyse, (Hodge et Austin, 2004), définit les anomalies comme « une observation qui semble incompatible avec le reste de l'ensemble de données ». Keogh *et al.* (2007) ont proposé une autre définition des anomalies, les discordes de séries temporelles, qui sont des sous-séquences différentes de toutes les sous-séquences réelles.

Le terme aberrant a été associé au bruit, reliant ces observations à des comportements incorrects ou incohérents (Aggarwal, 2015) ; par exemple, les erreurs humaines qui sont introduites lors de la récupération de données (Barai et Dey, 2017). Dans d'autres situations, la détection d'instances avec un écart élevé est considérée également comme des valeurs aberrantes. Selon Hawkins (1980), une valeur aberrante est une observation qui s'écarte fortement des autres observations. L'explication de ce phénomène peut être ramenée au fait qu'elle ait été générée par un mécanisme probablement différent.

Les termes anomalie et valeur aberrante donnent l'idée d'un modèle indésirable et dans notre travail nous utilisons les deux termes d'une manière interchangeable.

Il existe deux groupes de méthodes pour la détection d'anomalies (Däubener *et al.*, 2019) :

— Approches directes telles que le regroupement, la classification ou les méthodes basées sur la distance ou la densité. Ainsi, les anomalies sont considérées comme éloignées des centres des clusters, pour former une très petite classe/cluster à part entière, ou éloignées de leurs voisins les plus proches, ou avoir une distance probabiliste élevée.

— Approches indirectes (ou résiduelles) où le comportement normal est appris et modélisé. Sur la base de ces modèles, des prédictions sont faites et l'écart entre la valeur observée et la valeur prédite est utilisé pour décider si une observation est anormale.

En fonction de la méthode utilisée, la sortie d'un algorithme de détection d'anomalies peut être :

— Des scores : la plupart des algorithmes de détection des valeurs aberrantes produisent un score quantifiant le niveau d'aberrance de chaque point de données. Ce score peut également être utilisé pour classer les points de données par ordre de tendance aberrante. Il s'agit d'une forme de sortie très générale qui ne fournit pas un résumé concis du petit nombre des observations qui devraient être considérées

comme des valeurs aberrantes.

— Des labels binaires : c'est une étiquette indiquant si un point de données est une valeur aberrante ou non. Bien que certains algorithmes puissent renvoyer directement des étiquettes binaires, les scores aberrants peuvent également être convertis en étiquettes binaires. Ceci est généralement réalisé en imposant des seuils aux scores aberrants, et le seuil est choisi en fonction de la distribution statistique des scores. Ce type de sortie représente le résultat final qui est souvent nécessaire pour la prise de décision dans les applications pratiques.

Dans notre travail, nous utilisons une approche directe pour détecter les anomalies permettant de générer des labels décrivant les anomalies.

Pour conclure, la détection d'anomalies a suscité plusieurs travaux de recherche selon la nature des données, la disponibilité des labels sur la normalité et les domaines d'application qui sont divers. Nous allons aborder ces notions dans les parties qui suivent.

2.3 Domaines d'applications

De nombreuses recherches ont été effectuées sur la détection d'anomalies ces dernières années dans différents domaine d'application notamment la détection d'intrusions, la détection de fraudes, la détection d'anomalies dans l'imagerie médicale, la détection de dommages industriels, la détection d'anomalies dans les données de capteurs, le traitement d'image ou vidéo, etc (Chandola *et al.*, 2009; Mehrotra *et al.*, 2017) :

— Systèmes de détection d'intrusion. La détection d'intrusions fait référence à la détection d'activités malveillantes (effractions, pénétrations et autres formes d'abus informatique) sur les données qui sont collectées sur le trafic réseau ou d'autres actions de l'utilisateur, dans de nombreux systèmes informatiques. La détection de ces activités malveillantes est primordiale pour la sécurité informatique (Javaid *et al.*, 2016; Jadidi *et al.*, 2013).

— Détection de fraude. Il s'agit de la détection d'activités frauduleuses dans des organisations commerciales telles que les banques, les agences d'assurance, les sociétés de téléphonie mobile, le marché boursier, etc. Parmi ces activités, nous pouvons citer les fraudes par carte de crédit telle que l'utilisation non autorisée de la carte bancaire, les fraudes par téléphones mobiles tels qu'un volume élevé d'appels, les fraudes à l'assurance automobile telle que les réclamations non autorisées et illégales. Les organisations sont intéressées par la détection immédiate de ces fraudes

pour éviter des pertes économiques. Les données dans ce domaine sont généralement constituées d'enregistrements définis sur plusieurs variables (Ahmed *et al.*, 2016; Akoglu et Faloutsos, 2013).

— Diagnostic médical. Dans de nombreuses applications médicales, les données sont collectées à partir de divers appareils tels que les scans de tomographie par émission de positons (TEP). Ces données peuvent présenter des anomalies à cause d'un état anormal du patient, des erreurs d'instrumentation ou des erreurs d'enregistrement. Les données dans ce domaine peuvent être temporelles et spatiales (Ukil *et al.*, 2016; Hauskrecht *et al.*, 2007).

— Détection des dommages industriels : les anomalies sont liées à des défauts de composantes mécaniques telles que les moteurs, les turbines, le débit d'huile dans les pipelines ou d'autres composants mécaniques. Les données collectées dans ce domaine ont un aspect temporel (Othman et Eshames, 2012; Purarjomandlangrudi *et al.*, 2014).

— Les réseaux de capteurs. Les capteurs sont souvent utilisés pour suivre divers paramètres d'environnement et de localisation dans de nombreuses applications du monde réel. Les anomalies dans les données de capteurs font référence à des défauts de capteurs ou des événements (tels que des intrusions) imprévus (Rajasegarar *et al.*, 2008; Hayes et Capretz, 2014; Rabatel *et al.*, 2011; Chakrabarti *et al.*, 2016). Les données de capteurs peuvent être binaires, discrètes, continues, audio, vidéo, etc.

— Les applications de traitement d'images et vidéos. Les anomalies dans ce domaine correspondent à la détection de mouvements rares ou inconnus comme par exemple la détection d'anomalies dans les applications de vidéosurveillance, ou à des régions qui apparaissent anormales sur l'image statique comme par exemple l'analyse d'imagerie par satellite (Au *et al.*, 2006; Sabokrou *et al.*, 2018). Les données ont des caractéristiques spatiales et temporelles.

2.4 Type d'anomalies

Dans la littérature (Chandola *et al.*, 2009), il existe une classification générale des anomalies qui s'applique dans plusieurs domaines d'application et qui peut être divisée en trois principaux types : ponctuelle, collective et contextuelle.

2.4.1 Anomalies de point

Les anomalies de point (ou globales) correspondent à un point de données considéré comme valeur aberrante car il est suffisamment différent ou éloigné de l'ensemble des données. La figure I.2.1 reprend l'exemple d'une série temporelle de consommation énergétique d'un bâtiment. Une observation qui a une valeur très élevée (surconsommation) par rapport la fourchette habituelle de consommation d'un bâtiment présente une anomalie de point ou ponctuelle.

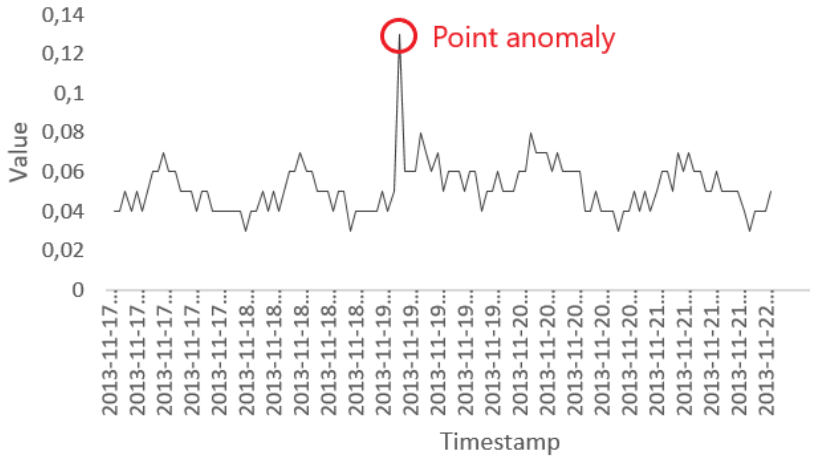


Figure I.2.1 – Anomalie ponctuelle dans une série temporelle de consommation énergétique d'un bâtiment.

2.4.2 Anomalies contextuelles

Les anomalies contextuelles (ou locales) correspondent à un point de données (ou une séquence de points) différent ou éloigné des autres points de données mais dans un contexte spécifique (spatial ou temporel). Par exemple, la figure I.2.2 présente une anomalie contextuelle dans une série temporelle de température mensuelle. Une basse température en hiver à l'instant t1 est considérée normale, tandis que le même cas pourrait ne pas être normal en plein été à l'instant t2.

2.4.3 Anomalies collectives

Les anomalies collectives (ou séquentielles) correspondent à une collection d'observations qui est différente de l'ensemble des données. Par exemple, la figure I.2.3 montre un exemple d'une série temporelle contenant une sous-série anormales parce qu'elle est différente par rapport à l'ensemble de sous-séquences de la série temporelle. Ceci peut correspondre à compteur en arrêt, qui échoue à remonter des données.

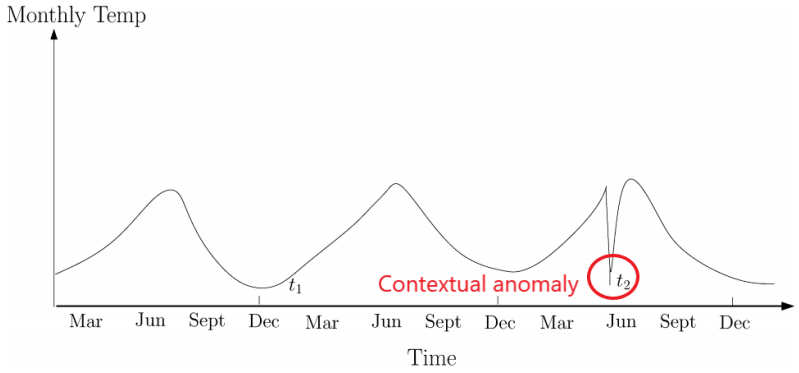


Figure I.2.2 – Anomalie contextuelle dans une série temporelle de température mensuelle (Chandola *et al.*, 2009).

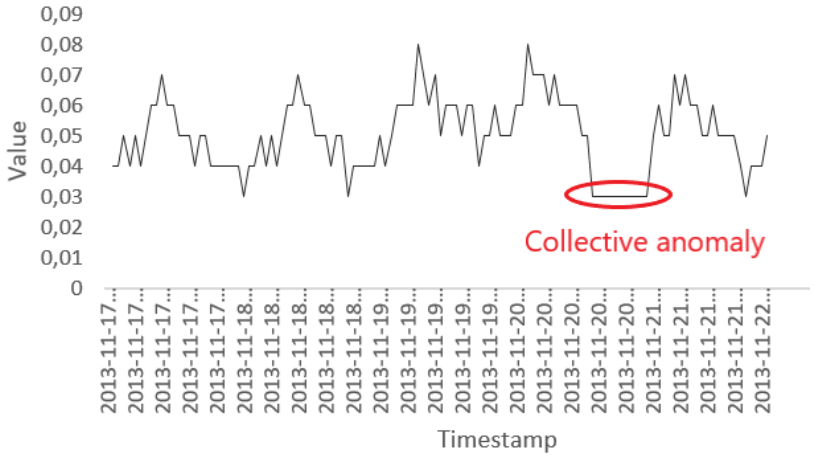


Figure I.2.3 – Anomalie collective correspondant à un arrêt de compteur.

2.4.4 Type d'anomalies dans les déploiements réels

À partir de l'état de l'art, nous avons construit le tableau I.2.1 qui montre l'équivalence entre les types d'anomalies observées dans des déploiements réels, que nous cherchons à détecter, et les types d'anomalies abordées dans la littérature. Comme le montre le tableau, il existe plusieurs terminologies pour identifier les types d'anomalies. Dans notre travail, nous cherchons les types d'anomalies dans les réseaux de capteurs, à savoir les anomalies engendrées par lectures défectueuses de capteurs (e.g., capteurs endommagés, changement de capteur) ou des évènements imprévus comme une coupure (e.g., problème de communication ou fausses alarmes). Ainsi, nous cherchons à détecter les anomalies suivantes :

— Pic positif ou négatif. C'est un changement brutal dans les lectures de capteurs mesuré entre deux échantillons successifs (représenté par des triangles dans la figure I.2.4. Nous pouvons l'associer comme une anomalie ponctuelle (globale)

Tableau I.2.1 – Comparaison entre les anomalies observées dans les déploiements réels et les anomalies détectées par les algorithmes de la littérature.

Références \ Anomalies	Pics	Bruit	Plateau	Changement de niveau
Chen et Liu (1993)	AO	TC	-	LS, SLS
Sharma *et al.* (2010)	courte	bruit	constante	-
Yao *et al.* (2010)	-	courte durée	longue durée	-
Upadhyaya et Singh (2012)	globale	locale	-	-
Yeh *et al.* (2016)	-	discord	-	-
Chen et Zhan (2008)	-	collective	-	-
Feremans *et al.* (2019)	-	contextuelle	-	-
Rosner (1983)	globale	locale	-	-
Basseville et Nikiforov (1993)	-	-	-	level shift

mais aussi comme une courte anomalie comme Sharma *et al.* (2010) l'ont définie ou encore comme étant AO (Additional Outlier) dans (Chen et Liu, 1993).

— Anomalies de bruit. Il s'agit d'une augmentation de la variance des lectures du capteur comme illustré dans la figure I.2.4. Contrairement aux anomalies courtes qui affectent un seul échantillon à la fois, les anomalies de bruit affectent un certain nombre d'échantillons successifs. Ce type d'anomalie présente les discordes de séries temporelles (Yeh *et al.*, 2016; Owuor *et al.*, 2018). Yao *et al.* (2010) ont défini ce type d'anomalie comme anomalie de courte durée dans les lectures du capteur. Nous pouvons les définir également comme anomalies locales (Rosner, 1983; Upadhyaya et Singh, 2012) ou comme changement temporaire (TC) (Chen et Liu, 1993).

— Anomalie constante (plateau) : le capteur signale une valeur constante pour un grand nombre d'échantillons successifs. Il s'agit des lectures anormales à un décalage constant (illustré par un rectangle dans I.2.4 4). (Yao *et al.*, 2010) ont défini ce type de valeur aberrante comme anomalie de longue durée à cause d'un changement relativement long dans les lectures des capteurs. Ce type d'anomalie peut être considéré comme anomalie collective (Chen et Zhan, 2008).

— Changement de niveau : c'est un changement brusque dans les mesures de capteurs, représenté par une croix dans la figure I.2.4, engendrant un changement de niveau permanent ou temporaire dans la série temporelle par une certaine amplitude à partir d'une observation (Balke, 1993). Ce type d'anomalie a été défini comme Level Shift dans (Basseville et Nikiforov, 1993) et divisé en deux variations dans Chen et Liu (1993) : les changements de niveau (Level Shift (LS)) et les changements de niveau saisonniers (Seasonal Level Shifts (SLS)) sont pris en

compte.

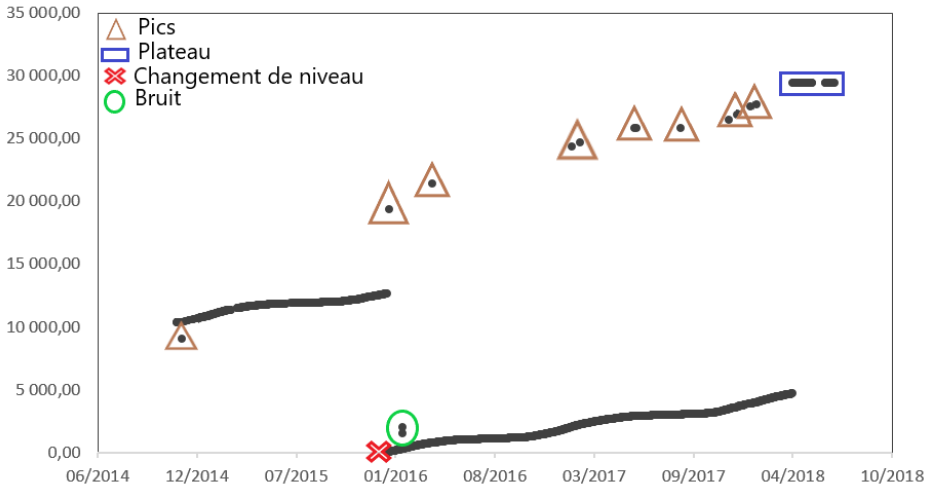


Figure I.2.4 – Exemple d'anomalies dans les mesures de capteurs.

2.5 Apprentissage automatique pour la détection d'anomalies

Les données disponibles influent sur les techniques de détection d'anomalies qui peuvent être appliquées. En effet, les instances de données peuvent être étiquetées (il existe une étiquette à chaque point de données donnant des informations si la classe de l'instance est normale ou anormale) ou non étiquetées. La détection peut alors se faire suivant les trois principes connus en apprentissage automatique à savoir : non supervisé, supervisé et semi-supervisé (Chandola *et al.*, 2009).

— Détection d'anomalies non supervisée : l'apprentissage non supervisé est utilisé dans le cas où nous ne disposons pas de données étiquetées. Cette approche permet de déterminer les valeurs aberrantes sans connaissances préalables des données. Les techniques qui fonctionnent en mode non supervisé ne nécessitent pas de données d'entraînement mais supposent que le comportement normal est le plus fréquent. L'avantage de cette méthode est qu'aucune donnée étiquetée n'est nécessaire, et elle est largement applicable dans différents domaines.

— Détection d'anomalies supervisée : cette approche nécessite un ensemble de données d'apprentissage qui contient des données étiquetées comme normales ou anormales. Le défi de l'apprentissage supervisé est qu'il est généralement très long d'étiqueter les données et qu'il est normalement difficile d'inclure tous les types

d'anomalies, ce qui est nécessaire pour que l'algorithme fonctionne correctement. Son avantage est qu'il peut être utilisé lorsque les anomalies sont plus fréquentes que les instances normales. Contrairement aux méthodes non-supervisées, les méthodes supervisées sont conçues pour la détection d'anomalies spécifiques à l'application.

— Détection d'anomalies semi-supervisée : cette approche suppose que les données d'apprentissage contiennent des instances partiellement étiquetées, par exemple pour seulement la classe normale. Comme le mode supervisé, il peut être difficile de trouver des données qui couvrent toutes les instances normales.

En règle générale, les méthodes non supervisées sont souvent utilisées dans un contexte exploratoire, où les valeurs aberrantes découvertes sont fournies à l'analyste pour un examen plus approfondi de leur importance spécifique à l'application. Tandis que les méthodes supervisées se font sur la base d'une vérité ; en d'autres termes, nous avons une connaissance préalable de ce que devraient être les valeurs de sortie de nos échantillons. Dans notre travail, nous traitons ces deux modes d'apprentissage.

2.6 Taxonomie des techniques de la détection d'anomalies

La plupart des recherches existantes portent soit sur plusieurs domaines d'applications, soit sur un seul domaine d'application comme le cas de ces revues (Hodge et Austin, 2004; Chandola *et al.*, 2009; Agrawal et Hori, 2015; Wu, 2016). Dans ces études, les auteurs ont discuté plusieurs techniques de détection d'anomalies selon le domaine d'application. Certains auteurs ont choisi les techniques qui sont appropriées pour détecter des types d'anomalies particulières (Sharma *et al.*, 2010). Ainsi, ces auteurs ont exploré les techniques de détection d'anomalies qui sont appropriées pour détecter les types d'anomalies (courte, bruit, et constante). D'autres présentent dans leur article une taxonomie pour les techniques de détection d'anomalies par rapport à plusieurs types de jeux de données (simple, complexe) (Zhang *et al.*, 2007). Xu *et al.* (2019) fournissent une étude plus récente des progrès réalisés dans la détection d'anomalies. Des revues récentes des méthodes de détection d'anomalies dans les séries temporelles univariées (Däubener *et al.*, 2019; Braei et Wagner, 2020) et multivariées (Blázquez-García *et al.*, 2020) ont été proposées.

Compte tenu de l'étendue de la littérature sur la détection des anomalies, nous avons regroupé les travaux en approches reposant sur les connaissances, la décomposi-

tion spectrale, la théorie de l'information, le clustering, la régression, la classification, l'exploration des motifs (pattern mining), les plus proches voisins et les statistiques comme illustré dans la figure I.2.5 (Chandola *et al.*, 2009; Omar *et al.*, 2013; Lakshmi *et al.*, 2020). Dans ce qui suit, nous donnons une brève explication des différentes techniques et nous nous concentrons sur les principales méthodes, en particulier celles que nous avons appliqué sur les données de capteurs.

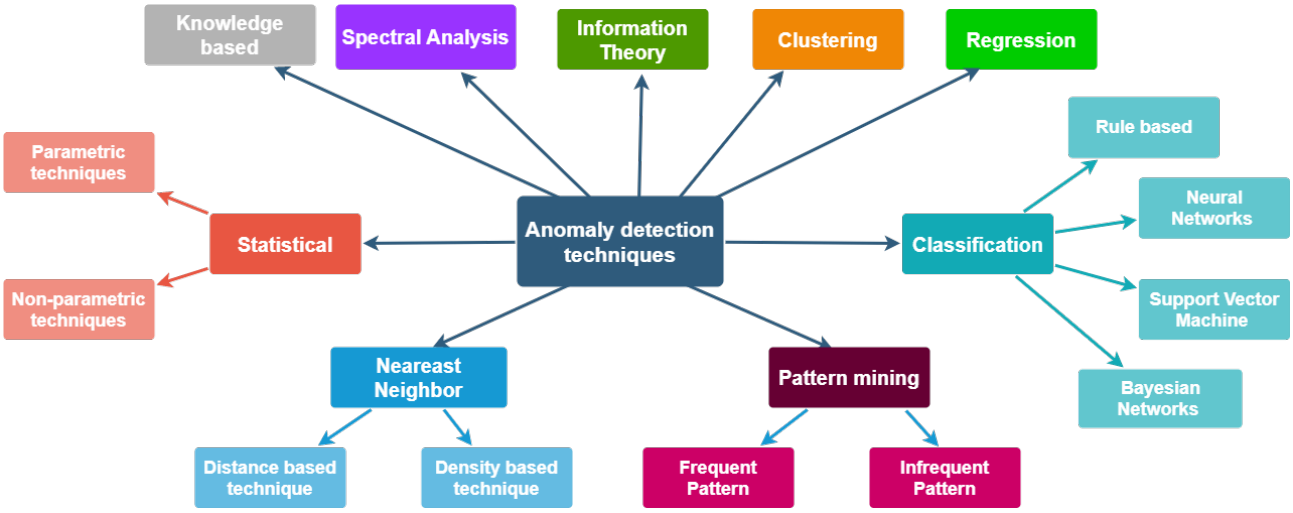


Figure I.2.5 – Les techniques de détection d'anomalies dans les données statiques.

2.6.1 Techniques basées sur les connaissances

Cette approche (également appelée système expert) est basée sur la connaissance humaine du domaine. Elle suppose que les motifs d'anomalies sont connus et qu'il est possible de définir ces modèles de manière compréhensible par la machine. Le fonctionnement d'un système basé sur la connaissance comporte trois étapes (Garcia-Teodoro *et al.*, 2009). Tout d'abord, un expert doit analyser manuellement une grande quantité de données et identifier les modèles d'anomalies. Ensuite, les modèles d'anomalies sont implémentés dans un système automatisé. Enfin, le système s'exécute automatiquement et détecte les anomalies dans les nouvelles données. Les connaissances d'experts peuvent être programmées de différentes manières (Sekar *et al.*, 2002; Estevez-Tapiador *et al.*, 2004; Zaki et Meira, 2014) :

— Systèmes basés sur des règles : contient un ensemble de règles « if-then » liées à différents événements. Il suppose que les données sont déterministes (les mêmes conditions conduisent toujours à la même conséquence). Il est simple à mettre en œuvre ; cependant, la liste des règles doit être exhaustive.

— Systèmes basés sur les statistiques : Il s'agit de créer des règles de décision basées sur de simples calculs statistiques. Par exemple, Sharma *et al.* (2010) ont défini deux règles selon le type d'anomalies à détecter. Une règle d'anomalie courte est définie pour traiter la série temporelle en comparant à chaque fois deux observations successives. Une anomalie est détectée si la différence entre ces observations est supérieure à un seuil donné. Une règle d'anomalie constante est aussi définie pour calculer l'écart-type sur un ensemble d'observations successives. L'ensemble est déclaré comme une anomalie si le résultat de l'écart type est égal à 0.

— Machine à états finis : une manière plus formelle d'encoder les connaissances d'experts. Une machine à états encode la succession d'événements de manière efficace. Cette méthode peut compacter un grand ensemble de règles dans un graphique simple qui facilite les tâches des experts.

2.6.2 Techniques basées sur les statistiques

Ces techniques adaptent un modèle statistique (généralement pour un comportement normal) aux instances données, puis appliquent un test d'inférence statistique pour déterminer si une nouvelle instance appartient ou non à ce modèle. Les instances qui ont une faible probabilité d'être générées à partir du modèle appris, en fonction de la statistique de test appliqué, sont déclarées comme des anomalies. Les approches basées sur les statistiques sont catégorisées en approches paramétriques et non paramétriques (Sreevidya, 2014).

— Les techniques paramétriques supposent la connaissance de la distribution sous-jacente et estiment les paramètres à partir des instances données.

— Les techniques non paramétriques ne supposent généralement pas la connaissance de la distribution sous-jacente et elles sont basées sur la construction du modèle de distribution.

Le test ESD généralisé (Extreme Studentized Deviate) (Rosner, 1983) et le changement de point (Basseville et Nikiforov, 1993; Aminikhanghahi et Hori, 2017) ont été proposés pour la détection d'anomalies sur des données unies-variées. L'algorithme ESD utilise des fonctions statistiques telles que la moyenne et la déviation standard pour la détection d'anomalies. ESD nécessite de spécifier une limite supérieure pour le nombre probable d'anomalies existantes ; ceci n'est pas possible pour toutes les applications. Hochenbaum *et al.* (2017) ont créé un algorithme nommé, Seasonal Hybrid ESD (SH-ESD) qui s'appuie sur le test ESD généralisé pour détecter les anomalies. SH-ESD peut être utilisé pour détecter les anomalies globales et locales. Ceci est réalisé

en employant la décomposition de séries chronologiques et en utilisant des métriques statistiques (médian et ESD).

La méthode de changement de point (Change Point) détecte les changements de distribution (e.g., moyenne, variance, covariance) dans les mesures du capteur (Rosner, 1983). Cette méthode détecte chaque changement sous forme d'anomalies.

2.6.3 Techniques basées sur la régression

Ces techniques sont largement utilisées sur les données temporelles. Cette approche est basée sur le principe de la prévision. Une anomalie peut être définie comme un écart entre la réalité et ce qui était attendu. Le principe de cette approche fonctionne dans le mode suivant. Premièrement, elle effectue une estimation des données à venir. Ensuite, elle quantifie l'écart entre la valeur réelle et la valeur prédite. Si l'écart est suffisamment grand (supérieur à un seuil prédéfini), le nouveau point de données est considéré comme une anomalie. Pour prévoir de nouvelles valeurs, nous avons besoin d'un modèle autorégressif tel que ARIMA (AutoRegressive Intergrated Moving Average) proposé par Chen et Liu (1993). De nombreux chercheurs ont appliqué ARIMA pour détecter des anomalies dans différents contextes (Moayedi et Masnadi-Shirazi, 2008; Zhu et Sastry, 2011; Pena *et al.*, 2013). Les modèles ARIMA sont connus pour être très précis dans les valeurs de prévision et extensibles aux séries chronologiques saisonnières. Il permet de détecter 5 types d'anomalies : AO (Additive Outlier), IO (innovation outilier),TC (Temporary Changes) ou LS (Level Shift), Seasonal Level Shift (SLS). Cependant, cette précision dépend fortement de la sélection de l'ordre du modèle (ordres auto-régressifs, différenciation et moyenne mobile). La principale faiblesse d'ARIMA est qu'il ne dispose pas d'une procédure efficace de mise à jour du modèle. Ceci rend ARIMA coûteux en calcul.

2.6.4 Techniques basées sur la classification

Les techniques de détection d'anomalies basées sur la classification fonctionnent en majorité dans un environnement supervisé ou semi-supervisé. Elles utilisent un ensemble d'apprentissage de données étiquetées (entrainement) pour apprendre un modèle ou un classificateur. Ce modèle est ensuite utilisé pour classer les nouveaux points (test) dans l'une des classes (normales, anormales). Comme illustré dans la figure I.2.5, les approches fondées sur la classification sont classées en quatre catégories : les approches basées sur les réseaux neurones (Ozyildirim et Avci, 2013), les approches basées sur les réseaux bayésiens (Rashidi *et al.*, 2011), les machines à vecteurs de support (Hejazi et

Singh, 2013) et les approches basées sur les règles (Duffield *et al.*, 2009), selon le type de modèle de classification qu'elles utilisent.

— Support Vector Machine (SVM) est un algorithme de classification largement utilisé dans les systèmes de détection d'anomalies, dans le cas des données numériques et de série temporelles (Ma et Perkins, 2003; Mukkamala *et al.*, 2002). SVM crée une frontière entre les données normales et anormales sur la base d'une fonction du noyau. SVM permet une précision de classification élevée lorsque des noyaux non linéaires (par exemple, polynôme) sont utilisés. Cependant, avec les noyaux non linéaires, SVM est très sensible aux problèmes de sur-ajustement.

— Des classificateurs de réseaux de neurones ont été utilisés dans le cadre de la détection d'intrusions (Javaid *et al.*, 2016; Jadidi *et al.*, 2013). D'autres chercheurs ont utilisé un algorithme de réseaux de neurones C-LSTM pour effectuer la détection d'anomalies dans les données de trafic Web (Kim et Cho, 2018). Ils ont combiné un réseau de neurones convolutifs (CNN), long short-term memory (LSTM) et deep neural network (DNN) pour modéliser les informations spatiales et temporelles contenues dans les données de trafic. Ils ont transformé le contexte temporel en utilisant une couche CNN. Ensuite, ils ont utilisé la sortie de cette couche CNN comme entrée pour plusieurs couches LSTM pour réduire les variations temporelles. La sortie de la couche LSTM finale est introduite dans plusieurs couches DNN entièrement connectées afin de classer la sortie. En conséquence, ils ont obtenu des performances de classification élevées pour les anomalies mais l'inconvénient des réseaux de neurones est leur nature « black box ». En effet, l'humain n'a pas de contrôle sur les règles de décision apprises et les algorithmes appris sont difficilement interprétables.

— Les Réseaux Bayésiens sont un formalisme qui fait la fusion entre la théorie de probabilités et la théorie de graphe. Ils sont constitués d'un ensemble de variables (nœud du réseau) et d'un ensemble d'arcs entre les variables. Dans cette technique, la structure bayésienne est apprise à partir des données et les paramètres du réseau bayésien sont estimés. Cet algorithme a été utilisé dans (Petkovic *et al.*, 2002) pour extraire les moments intéressants dans des vidéos de Formule 1. Cette approche utilise une structure du réseau bayésien construite à la main, à partir de connaissances sur le domaine traité. Le problème est que les connaissances sur les différentes relations existantes entre les variables ne sont pas toujours disponibles. Dans (Rashidi *et al.*, 2011), les auteurs ont présenté une méthode pour trouver des anomalies dans des ensembles de données catégorielles ou mixtes de manière non supervisée.

Cette technique est incapable de détecter ou d'expliquer des anomalies plus complexes résultant de conflits entre de grands ensembles de nœuds plutôt que des nœuds individuels.

— Les techniques basées sur les règles se concentrent sur l'apprentissage des règles qui capturent l'état normal du système. L'avantage le plus évident de l'adoption d'une approche basée sur des règles est que ces dernières représentent les raisonnements du monde réel expliquant pourquoi une valeur est anormale tels que les arbres de décision (Gaddam *et al.*, 2007; Muniyandi *et al.*, 2012; Gupta *et al.*, 2017; Sinwar et Kumar, 2016).
En général, les systèmes basés sur des arbres de décision sont facilement compréhensibles car ils montrent clairement la succession de tests conduisant à classer un point comme normal ou anormal comme illustré dans la figure I.2.6. Son modèle appelé « white box » est facile à conceptualiser, visualiser et interpréter le résultat, et permet également de générer des règles de décision compréhensibles (voir figure I.2.6). Bien que les approches fondées sur des règles sont rapides à tester, cela peut ne pas toujours être le cas lorsqu'il existe un ensemble de règles vaste et complexe et ceci rend les règles générées moins interprétables. Plusieurs classificateurs basés sur l'arbre de décision ont été explorés dans (Sinwar et Kumar, 2016) pour la détection d'anomalies tels que Best-first, Decision Tree, FunctionalTree, Logistic Model Tree, J48 et Random Forest. Sur la base de leur étude, ils ont montré que l'arbre de décision Random Forest a surpassé les autres classificateurs basés sur l'arbre de décision en terme de taux de classification correct.

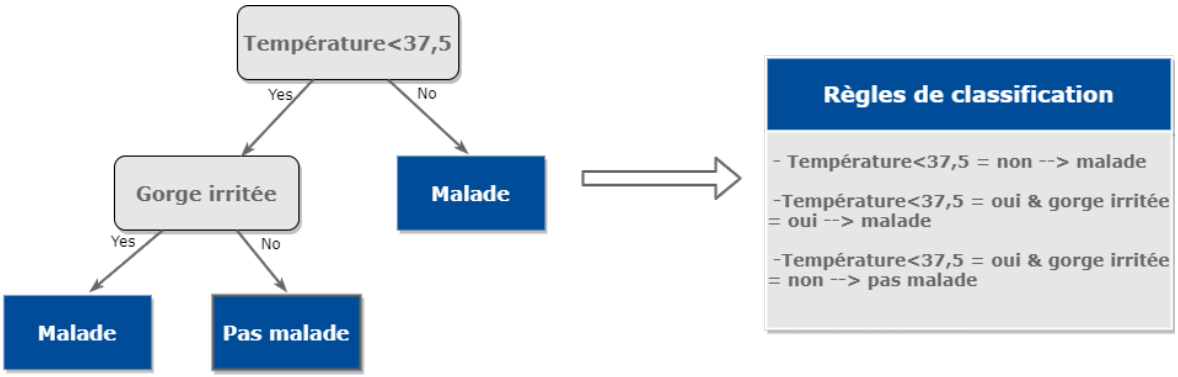


Figure I.2.6 – Technique de classification basée sur des règles d'arbre de décision.

Le principal problème des arbres de décision est qu'en réduisant un problème de classification complexe avec des données hautement multidimensionnelles, le risque de sur-ajustement est élevé. Pour contourner ce problème, les forêts aléatoires ont été proposé par Breiman (2001). Une forêt aléatoire consiste à créer plusieurs arbres de décisions, chacun sur un sous-ensemble sélectionné au hasard à partir de

l'ensemble de données d'entrée. La sortie d'une forêt aléatoire est la moyenne de tous les arbres de décision (Chen *et al.*, 2004).

2.6.5 Techniques basées sur l'exploration de motifs

Nous pouvons classer ces approches en 2 catégories : l'exploration de modèles fréquents et l'exploration de modèles rares.

— L'exploration de modèles ou motifs fréquents fait référence à la tâche d'extraction de modèles informatifs et utiles dans des ensembles de données. Le but est de trouver des modèles fréquents dans les données, servant de modèle pour le comportement normal fréquemment observé (Kuchar et Svátek, 2018; Abghari *et al.*, 2018). La tâche d'exploiter des modèles fréquents apparaît dans de nombreux domaines. Une application typique est l'analyse du panier de marché, dont l'objectif est d'extraire les ensembles d'articles qui sont fréquemment achetés ensemble dans un supermarché en analysant les paniers des clients. Une fois que nous extrayons les ensembles fréquents, ils nous permettent d'extraire des règles d'association parmi les ensembles d'éléments, où nous faisons une déclaration sur la probabilité que deux ensembles d'éléments se reproduisent ou se produisent conditionnellement (Zaki et Meira, 2014). De nombreux algorithmes ont utilisé l'extraction de motifs fréquents pour la détection d'anomalies tel ques PBAD (Pattern-Based method for Anomaly Detection) (Feremans *et al.*, 2019), POD (Pattern based Outlier Detection) (Zhang et Jin, 2010), Fp-outlier (Frequent pattern based outlier detection) (He *et al.*, 2005) et MP (Matrix Profile) proposé par Yeh *et al.* (2016).

— L'exploration de modèles peu fréquents a attiré l'attention de la communauté de recherche sur l'exploration de données qui vise à découvrir des associations rares parce que les modèles peu fréquents sont plus intéressants que les modèles fréquents dans la détection d'anomalies (Ghoting *et al.*, 2004; Chen et Zhan, 2008; Bouasker et Ben Yahia, 2015; Rahman *et al.*, 2016; Yeh *et al.*, 2016; Owuor *et al.*, 2020) ou de la nouveauté. Un algorithme de détection d'anomalies multi-échelles (PAV) basé sur des modèles linéaires peu fréquents a été proposé dans (Chen et Zhan, 2008). Les modèles d'anomalies sont des modèles peu fréquents avec un support inférieur à celui des autres modèles de séries chronologiques.

2.6.6 Techniques basées sur les plus proches voisins

Ces approches sont classées en deux catégories : les techniques qui utilisent la distance d'une instance de données à son $k^{ème}$ voisin le plus proche comme score d'ano-

malie (Upadhyaya et Singh, 2012) et les techniques qui calculent la densité relative de chaque instance de données pour calculer son score d'anomalies, par exemple, l'algorithme LOF (Local Outlier Factor) (Breunig *et al.*, 2000). La méthode de détection des valeurs aberrantes basée sur la densité, estiment la densité du voisinage de chaque instance de données. Une instance située dans un voisinage à faible densité est déclarée anormale tandis qu'une instance située dans un voisinage dense est déclarée normale. Breunig *et al.* (2000) ont proposé l'algorithme LOF (Local Outlier Factor). Dans cette approche, une valeur aberrante est mesurée en utilisant un facteur de valeur aberrante locale (LOF), qui est le rapport entre la densité locale de ce point et la densité locale de son voisin le plus proche. Le point de données dont la valeur LOF est élevée est déclaré comme aberrant.

2.6.7 Techniques basées sur le partitionnement

La méthode basée sur le partitionnement regroupe des instances de données similaires pour former des clusters. Le partitionnement est une technique non supervisée ou semi supervisée. Les techniques de détection d'anomalies basées sur le partitionnement peuvent être regroupées en trois catégories qui reposent sur les hypothèses suivantes :

— Les instances de données normales appartiennent à un cluster dans les données, tandis que les anomalies n'appartiennent à aucun cluster.

— Les instances de données normales se trouvent à proximité du centre de gravité du cluster le plus proche, tandis que les anomalies sont loin du centre de gravité du cluster le plus proche.

— Les instances de données normales appartiennent à des clusters volumineux et denses, tandis que les anomalies appartiennent à des clusters petits ou clairsemés.

De nombreux algorithmes de clustering ont été utilisés pour détecter des anomalies (Hardin et Rocke, 2004) tels que DBSCAN, ROCK, K-Means etc . L'un des algorithmes les plus utilisés est K-means (Gaddam *et al.*, 2007; Münz *et al.*, 2007; Muniyandi *et al.*, 2012). Le principe de l'algorithme K-means est comme suit : on commence par sélectionner les centroïdes et créer des clusters autour d'eux en affectant chaque point de données à son centroïde le plus proche. Ensuite, les centres de gravité et les clusters sont mis à jour à plusieurs reprises jusqu'à la convergence. Gançarski *et al.* (2020) proposent une plateforme, FODOMUST, de clustering collaboratif sous contraintes incrémental de séries temporelles. Elle contient des méthodes,librairies et interfaces dédiées au clustering de données complexes. Bien que k-means soit simple à mettre en œuvre et facile à interpréter, il présente quelques défauts comme le nombre de clusters

qu'il faut choisir avant d'exécuter l'algorithme et surtout la forme sphérique des clusters qu'il suppose (El Malki *et al.*, 2020).

2.6.8 Techniques basées sur la théorie d'information

Les techniques de théorie de l'information analysent le contenu informationnel d'un ensemble de données en utilisant différentes mesures théoriques de l'information telles que la complexité de Kolomogorov, l'entropie, l'entropie relative, etc. Cette approche est basée sur l'hypothèse que les anomalies dans un ensemble de données modifient son contenu d'information. L'indicateur d'information le plus utilisé en théorie de l'information est l'entropie qui quantifie l'incertitude ou le caractère aléatoire des données. L'entropie a été utilisée par de nombreux chercheurs pour détecter des anomalies (Nychis *et al.*, 2008; Bereziński *et al.*, 2015), mais le problème avec cette technique est sa grande sensibilité à la présence de bruit.

2.6.9 Techniques basée sur l'analyse spectrale

Les techniques spectrales tentent de trouver une approximation des données en utilisant une combinaison d'attributs qui capturent l'essentiel de la variabilité des données. Cette approche suppose que les points normaux et anormaux sont facilement séparables dans un espace dimensionnel inférieur. Ainsi, nous projetons les données dans cet espace. Parmi les techniques d'analyse spectrale nous pouvons citer, l'analyse en composantes principales (PCA) (Ringberg *et al.*, 2007; Harrou *et al.*, 2015), la transformation de Fourier rapide (FFT) (Han *et al.*, 2014), la transformée en ondelettes (Mallat, 2000) et transformée de Hough (Lu et Ghorbani, 2008; Du *et al.*, 2018).

2.7 Méthodes d'évaluation

Dans cette section, nous donnons un aperçu des méthodes d'évaluation utilisées pour évaluer les performances d'un algorithme de détection d'anomalies. La comparaison et l'évaluation des résultats des méthodes de détection d'anomalies sont basées sur l'analyse des observations ayant été détectées à tort ou à raison comme des anomalies ou comme des comportements normaux.

2.7.1 Matrice de confusion

Une matrice de confusion est un tableau qui rassemble les résultats obtenus de l'application d'un algorithme d'apprentissage automatique (supervisé ou non supervisé). Dans le cas d'une détection d'anomalies, nous avons quatre grandeurs dans la matrice de confusion :

Tableau I.2.2 – Matrice de confusion

		Prédit	
		Positive	Négative
Réel	Positive	VP	FN
	Négative	FP	VN

Les comportements qui sont qualifiés dans nos travaux de Négatifs sont les comportements normaux et de Positifs pour les cas d'anomalies.

— Vrai négatif (VN) : une valeur normale étiquetée correctement par l'algorithme.

— Vrai positif (VP) : une anomalie qui est correctement prédite/classée par l'algorithme.

— Faux positif (FP) : une valeur normale, considérée à tort comme une anomalie.

— Faux négatif (FN) : une anomalie étiquetée, considérée à tort comme valeur normale.

2.7.2 Métrique d'évaluation

Plusieurs métriques peuvent être déduites de la matrice de confusion. Parmi celles qui sont utilisées fréquemment nous trouvons la précision (également appelée valeur prédictive positive), le rappel (également connu sous le nom de sensibilité) et F-mesure qui est un compromis (moyenne harmonique) entre le rappel et la précision (Sokolova *et al.*, 2006).

— La précision : quantifie la pertinence des points détectés comme anomalies. Elle mesure la probabilité qu'une observation classée Positif soit effectivement Positif.

$$P = VP/(VP + FP) \tag{2.1}$$

— Le rappel : quantifie la capacité d'un algorithme à détecter les anomalies existantes. Il représente le taux de Vrais Positifs, c'est-à-dire la proportion d'anomalies correctement classées.

$$R = VP/(VP + FN) \tag{2.2}$$

— F-mesure (ou F1 Moyenne harmonique) de la précision et du rappel donnant la performance de l'algorithme

$$F - mesure = 2PR/(P + R) \tag{2.3}$$

Ces métriques sont les plus utilisées pour évaluer les algorithmes de classification des séries chronologiques (Tatbul *et al.*, 2018). Dans notre thèse, nous utilisons ces mesures pour évaluer nos approches de détection d'anomalies. Nous évaluons notre système en vérifiant a posteriori si les classes affectées aux observations sont les bonnes.

Tableau I.2.3 – Comparaison des approches de détection d'anomalies.

Techniques	**Algorithme**	**Anomalies**	**Entrée**	**Sortie**	**Apprentissage**	**Interprétable**	**Référence**
Les connaissances	Règles	courte, constante	time-series, numérique	labels	non supervisé	oui	Sharma *et al.* (2010)
Les plus proches voisins	Densité (LOF)	globale, locale	numérique	score	non supervisé	non	Upadhyaya et Singh (2012)
	Distance (KNN)	globale	numérique				Breunig *et al.* (2000)
Statistiques	SH-ESD	globale, locale	time-series	labels	non supervisé	oui	Hochenbaum *et al.* (2017)
	Change Point	level shift	time-series	label	non supervisé	oui	Basseville et Nikiforov (1993)
Regression	ARIMA	AO, TC, LS	time-series	label	non supervisé	oui	Chen et Liu (1993)
Classification	Arbre de décision	locale, globale	numérique/ catégorique	classes	supervisé	oui	(Sinwar et Kumar, 2016)
	Forét aléatoire					non	Breiman (2001)
	Réseau de nouronnes					non	Sreevidya (2014)
Exploration de motifs	PBAD	contextuelle	time-series	score	supervisé	non	Feremans *et al.* (2019)
	PAV	collective				non	Chen et Zhan (2008)
	MP	discordes				non	Yeh *et al.* (2016)
Exploration de motifs	**CoRP**	**pics, bruit, constant, changement niveau**	**time-series**	**label, type**	**non supervisé**	**oui**	**Ben Kraiem *et al.* (2019)**

2.8 Synthèse

Dans le tableau I.2.3, nous présentons les travaux de recherche de l'état de l'art destinés à résoudre le problème de la détection d'anomalies. Nous comparons ces travaux selon les critères suivants :

— la technique utilisée pour chaque algorithme de détection ;

— le ou les types d'anomalies pris en considération par chaque méthode ;

— l'entrée de chaque méthode qui correspond au type de données traitées ;

— la sortie décrivant les anomalies pouvant être labels, scores ou classes ;

— le type d'apprentissage entre supervisé et non-supervisé selon la disponibilité des étiquettes ;

— l'interprétablité qui indique si le résultat de chaque méthode est compréhensible par les experts.

Comme montre le tableau I.2.3, nous pouvons constater que certains algorithmes traitent les données de manière simple et donnent des résultats interprétables à travers des règles ou encore des labels indiquant avec précision l'anomalie et d'autres sont plus complexes et moins facilement interprétables quand il s'agit par exemple d'une boîte noire ou encore lorsque le résultat est un score qui demande un travail pour pouvoir interpréter le résultat. De plus, toutes ces approches traitent quelques types d'anomalies spécifiques mais ne couvrent pas tous les types d'anomalies que nous pouvons observer dans les déploiements réels.

Notre travail (dernière ligne du tableau I.2.3) permet de palier ces lacunes à travers une méthode, CoRP, qui permet de détecter finement de multiples anomalies de différents types et de générer comme résultat le label et le type d'anomalie trouvée. Nous proposons également une méthode d'étiquetage automatique des séries temporelles. Cette méthode, basée sur les motifs, est interprétable par les experts étant donné que les motifs décrivent les points remarquables et donc des anomalies potentielles.

2.9 Conclusion

Dans ce chapitre, nous avons présenté les principaux concepts nécessaires à la compréhension du contexte et de la problématique que nous abordons dans cette thèse. Nous avons présenté les principes de la détection d'anomalies, en commençant par la définition et les types d'anomalies en précisant les types d'anomalies spécifiques à notre

domaine d'application, les domaines d'application, puis le type d'apprentissage, les différentes techniques de détection et les méthodes d'évaluation pour comparer et évaluer les méthodes de détection d'anomalies. Nous avons résumé l'ensemble des travaux et nous avons construit un tableau synthétique où nous comparons les différents travaux de la littérature par rapport à nos travaux.
Dans la partie suivante, nous allons présenter nos contributions pour la détection d'anomalies.

Deuxième partie

Méthodes basées sur les motifs pour la détection d'anomalies

1 Introduction

« Ce qui est affirmé sans preuve peut être nié sans preuve. »

Euclide de Mégare (v. 450 av. J.-C. — v. 380 av. J.-C.)

Table des matières

Cette deuxième partie du mémoire présente les approches de détection d'anomalies que nous avons proposées. Pour ce faire, nous commençons par présenter le contexte et la motivation de notre stratégie ainsi que les notions de base utilisées dans notre travail avant de détailler les contributions proposées. Nous présentons ensuite notre première approche, basée sur les motifs, pour la détection d'anomalies multiples. Puis nous détaillons notre deuxième approche supervisée, d'extraction de règles pour la détection d'anomalies.

1.1 Contexte et motivation

Les réseaux de capteurs jouent un rôle important dans la supervision et l'exploration des réseaux de distribution de fluides (e.g., énergie, eau, chauffage) à l'échelle d'un campus et plus largement d'une ville, d'une région ou d'un pays. L'exploitation de ces réseaux repose sur des données relevées par des capteurs. Ces données comportent des anomalies qui nuisent à la supervision (e.g., fausses alarmes, arrêts) telles que les anomalies illustrées dans la figure II.1.1. Notre travail se place dans le cadre d'applications réelles ayant des anomalies spécifiques au métier à savoir la gestion des fluides sur le campus de Rangueil-Toulouse géré par le Service de Gestion et d'Exploitation (SGE). Les consommations énergétiques des bâtiments montrent parfois des écarts importants entre la demande d'énergie prévue et la consommation d'énergie réelle lors de l'exploitation du bâtiment. Ces problématiques demandent une phase d'analyse par les experts et une correction des consommations énergétiques avant d'établir les factures aux clients. Le SGE cherche à terme d'améliorer la stratégie de performance énergétique et le processus d'analyse en mettant en place un système de détection d'anomalie à posteriori. Notre objectif ainsi est de traiter les données issues des capteurs du SGE et de trouver une méthode permettant de détecter de multiples anomalies de différents types observés lors de déploiements réels tout en maximisant le nombre d'anomalies détectées et minimisant les erreurs de détections (FP, FN). Nous traitons dans ce contexte des séries temporelles uni-variées a posteriori.

En situation d'exploitation réelle, la supervision des réseaux de capteurs se fait par les experts (ingénieurs d'exploitation, techniciens de maintenance, etc.) en observant les courbes afin de détecter des points remarquables qui correspondent à des comportements inhabituels. Typiquement, ce sont les mesures des capteurs de notre étude de cas illustrées figure II.1.1. Ces points remarquables sont les variations inhabituelles entre les points successifs d'une série temporelle ; ils constituent les marqueurs (ou indices) de possibles anomalies. A travers la connaissance de l'historique et de la nature des

données, les experts analysent le voisinage des points remarquables afin de décider les instances qui représentent une vraie anomalie.

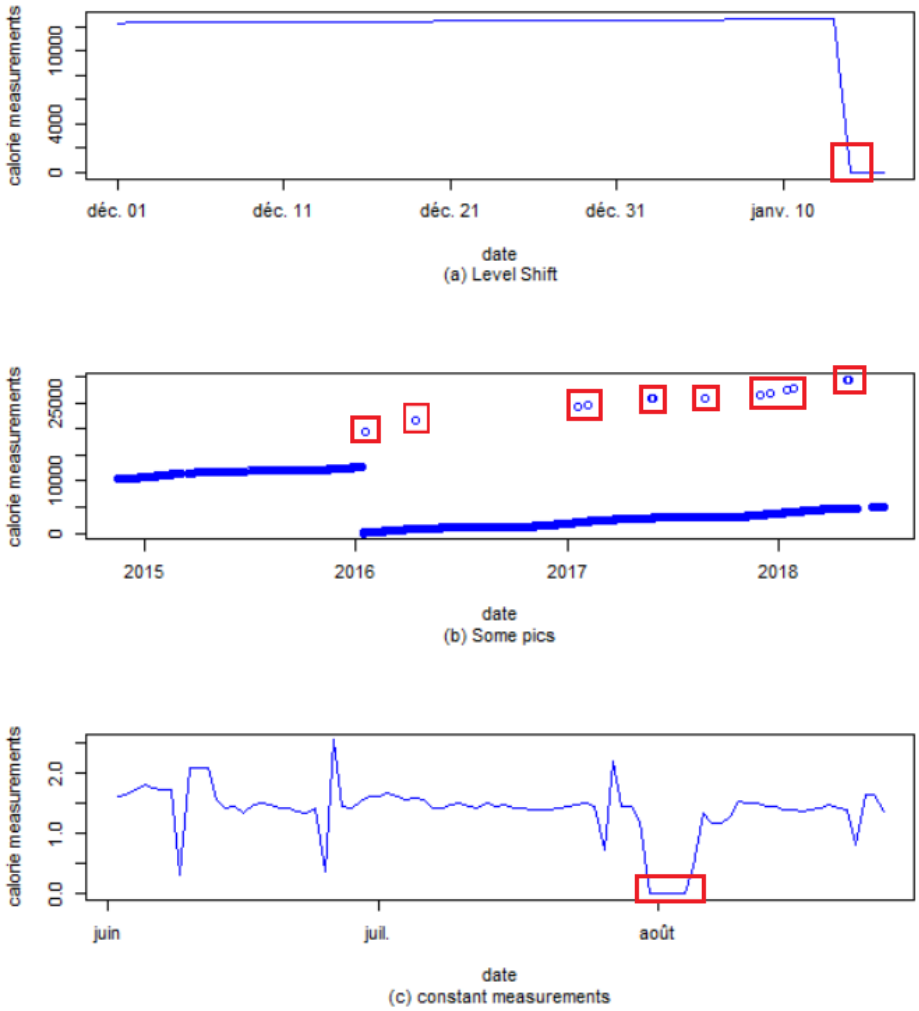


Figure II.1.1 – Exemple d'anomalies observées dans des déploiements réels.

Notre objectif est d'automatiser ce processus pour détecter des anomalies en premier lieu et pour générer des règles de classification interprétables par les experts en deuxième lieu.

1.2 Types d'anomalies

Comme nous l'avons vu dans le chapitre I.2, le choix d'une technique pour la détection d'anomalies est notamment fait en fonction des données traitées. Dans le contexte de notre travail, nous choisissons dans nos travaux d'appliquer la détection d'anomalies sur les données de monitoring du système de supervision géré par le SGE.

Par exemple, la figure II.1.1 (a) illustre un changement brusque dans les mesures de capteurs, engendrant un changement de niveau permanent suite à un problème du matériel (e.g., capteurs endommagés, changement de capteur). La figure II.1.1 (b) illustrent plusieurs pics représentant des défauts de lecture liés à un événement imprévu (e.g., panne, rupture). Enfin, la figure II.1.1 (c) représente un décalage constant dans les mesures (dû à un problème de communication).

Nous cherchons 4 types d'anomalies observées dans les déploiements réels comme illustré dans le tableau I.2.1. Ainsi, une anomalie peut se manifester par :

— Un décalage ou un changement de niveau dans les mesures de capteurs, associé au dépassement d'un seuil de valeur, suite à un problème de matériel (e.g., capteurs endommagés, changement de capteur). Il s'agit d'un décalage important entre les valeurs précédentes et les valeurs qui suivent ce changement, généré en raison d'un événement affectant une série à un moment donné.

— Un changement anormal ou une fréquence de variation inattendue dans les mesures de capteurs ou pics représentant des défauts de lecture liés à un événement imprévu (e.g., panne, rupture). Nous cherchons dans ce cas 2 types d'anomalies à savoir un pic positif et un pic négatif avec une forte variation par rapport aux autres instances de données ou une petite variation.

— Une variation brusque ou un bruit dans les mesures de capteurs liés à un événement imprévu (e.g., panne, rupture). Cette variation peut affecter plusieurs observations successives.

— Une valeur constante ou un décalage constant dans les mesures dû à un problème de communication. Ces valeurs constantes pourraient être une anomalie pour un nombre d'échantillons successifs avec ou sans décalage par rapport aux valeurs précédentes ;

1.3 Notations utilisées

Dans cette partie, nous présentons les définitions et notions de base utilisées dans nos approches.

Définition 1.II.1 *Une série temporelle* uni-variée, est composée d'observations ou de points successifs collectés séquentiellement dans le temps à intervalle régulier. Ces points représentent les mesures associées à un horodatage indiquant l'heure de sa collecte. Une *série temporelle* est définie comme $Ts = \{x_1, ..., x_n\}$ où $\forall i \in [1..n], x_i \in \mathbb{R}$ tels que les valeurs x_i sont uniformément espacées dans le temps et n est la taille de Ts.

Définition 2.II.1 *Un point (ou mesure)* est composé d'une valeur et d'un horodatage. On note un point $x_i = (t_i, v_i)$ tel que t_i est l'horodatage de x_i (noté $t(x_i)$) et v_i est la valeur de x_i (notée $v(x_i)$).

2 CoRP : Composition of Remarkable Points

« La meilleure façon de prédire l'avenir est d'étudier le passé, ou de pronostiquer. »

Ralph H. Kiyosaki (1919 — 1991)

Table des matières

2.1 Introduction

Pour superviser les données issues des capteurs, les experts analysent les courbes et détectent des points remarquables dans les séries chronologiques qui peuvent être considérés comme des modèles intéressants permettant de détecter des valeurs aberrantes de différents types.

Dans ce chapitre, nous présentons notre approche basée sur les motifs pour détecter des anomalies multiples pouvant survenir dans une série temporelle. Elle permet d'identifier finement différents types d'anomalies en précisant la localisation et le type d'anomalie.

2.2 Contexte et motivation

Plusieurs techniques de détection d'anomalies ont été proposées dans la littérature et classées selon les domaines d'applications ou les types d'anomalies à détecter (Chandola *et al.*, 2009). Néanmoins, ces techniques ne permettent pas toujours de détecter tous les types d'anomalies, obligeant les applications à utiliser plusieurs méthodes pour détecter des anomalies de natures diverses. En effet, les travaux existants ont du mal à détecter différents types d'anomalies et présentent encore de nombreux inconvénients lorsqu'ils sont appliqués aux anomalies multiples (Sharma *et al.*, 2010; Yao *et al.*, 2010; Munir *et al.*, 2017; Kiani *et al.*, 2020). La difficulté de disposer d'une technique robuste pour détecter l'ensemble des anomalies nous amène à définir une nouvelle méthode configurable nommée **CoRP** "Composition of Remarkable Points". Cette méthode permet, premièrement, de détecter des points qui paraissent remarquables dans les séries temporelles en évaluant des motifs et, deuxièmement, d'identifier de multiples anomalies en utilisant des compositions de points remarquables. CoRP nécessite l'expertise du domaine d'application pour pouvoir définir efficacement les motifs et les compositions de labels.

2.3 Description de CoRP

Dans cette thèse, nous avons travaillé avec les experts du Service de Gestion et d'exploitation sur le traitement et l'analyse des données. Ils nous ont montré leur stratégie d'analyse pour superviser les réseaux de capteurs. En effet, ils analysent les courbes issues de différents capteurs et détectent les points qui sont remarquables. Ces points sont dits remarquables parce qu'ils ont des comportements inhabituels

par rapports au reste des données. Ces points représentent des variations inhabituelles entre certains points successifs d'une série chronologique. Ensuite, ils analysent l'écart entre ces points. L'écart est très important pour eux, surtout pour les données d'index (relevé de compteurs) qui représentent des séries temporelles croissantes. Enfin ils analysent le voisinage de chaque point remarquable pour décider s'il représente une anomalie. Cette stratégie est manuelle mais elle représente une expertise et un savoir-faire des experts qui arrivent en inspectant les courbes à détecter différents types d'anomalies et d'analyser leurs causes.

Dans ce contexte, en s'appuyant sur l'expérience des experts (détection des points remarquables puis identification des anomalies), nous avons proposé l'algorithme CoRP construit en deux phases. La première consiste à détecter et annoter les points considérés comme remarquables dans la série temporelle. La deuxième phase consiste à identifier les anomalies à partir des compositions de points remarquables.

Il est important de noter que CoRP est générique ; il est donc facilement adaptable à d'autres situations en fournissant le cas échéant différents motifs.

2.3.1 Détection des points remarquables

La détection des points remarquables est réalisée à partir des motifs (motifs) de détection définis avec l'aide des experts.

Définition 1.II.2 *Un motif* p est défini par un triplet $p = (l, \sigma_a, \sigma_b)$ où l, est un label du point remarquable, σ_a et σ_b sont deux seuils utilisés pour décider si un point donné est remarquable. Un motif est appliqué sur trois points consécutifs x_{i-1}, x_i, x_{i+1} d'une série temporelle Ts.
σ_a correspond à l'écart entre $v(x_{i-1})$ et $v(x_i)$ tandis que σ_b correspond à l'écart entre $v(x_i)$ et $v(x_{i+1})$ comme illustré figure II.2.1. Lorsqu'un motif est vérifié sur x_{i-1}, x_i, x_{i+1}, le point x_i prend le label l du motif p.

Dans le cas échéant, où aucun motif n'est vérifié, le point x_i est considéré comme étant normal et étiqueté par le label « Normal ».

Définition 2.II.2 *Une série temporelle labellisée* Ts_L est une série temporelle de points sur lesquels sont ajoutés les labels détectés par les motifs.

Définition 3.II.2 *Un point remarquable* x_i d'une série temporelle labellisée est défini par un triplet (t_i,v_i, L_i) où t_i est son horodatage, v_i est sa valeur et $L_i = \{l_1, l_2, ...\}$ est une liste de labels caractérisant le point.

Les motifs sont donc utilisés pour détecter les points remarquables et leur associer

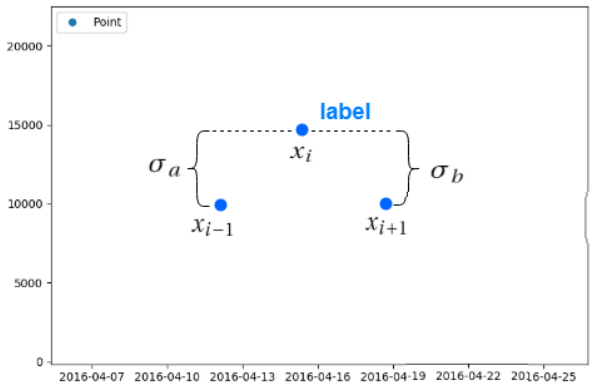
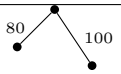
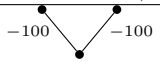
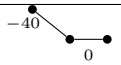
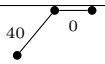
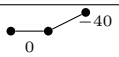
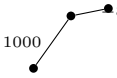
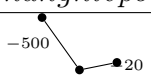


Figure II.2.1 – Étiquetage d'un point remarquable "x" par un motif σ_a et σ_b.

un ou plusieurs labels correspondants. Ainsi, la liste des labels d'un point remarquable est constituée des labels des différents motifs vérifiés sur ce point.

Tableau II.2.1 – Les différents motifs pour étiqueter les points remarquables dans une série temporelle.

Motif	Définition	Exemple
Ptpicpos	$x_{i-1} < x_i \wedge x_i > x_{i+1}$	
	$\sigma_a, \sigma_b \in R_+^*$	$Ptppicpos_{80,100}$
Ptpicneg	$x_{i-1} > x_i \wedge x_i < x_{i+1}$	
	$\sigma_a, \sigma_b \in R_-^*$	$Ptpicneg_{-100,-100}$
SartCstNeg	$x_{i-1} > x_i \wedge x_i = x_{i+1}$	
	$\sigma_a \in R_-^*, \sigma_b = 0$	$SartCstNeg_{-40,0}$
StartCstPos	$x_{i-1} < x_i \wedge x_i = x_{i+1}$	
	$\sigma_a \in R_+^*, \sigma_b = 0$	$StartCstPos_{40,0}$
EndCstNeg	$x_{i-1} = x_i \wedge x_i > x_{i+1}$	
	$\sigma_a = 0, \sigma_b \in R_+^*$	$EndCstNeg_{0,-40}$
EndCstPos	$x_{i-1} = x_i \wedge x_i < x_{i+1}$	
	$\sigma_a = 0, \sigma_b \in R_-^*$	$EndCstPos_{0,-40}$
CST	$x_{i-1} = x_i \wedge x_i = x_{i+1}$	
	$\sigma_a = 0, \sigma_b = 0$	$CST_{0,0}$
Changnivpos	$x_{i-1} < x_i \wedge x_i < x_{i+1}$	
	$\sigma_a \in R_+^* \sigma_b \in R_-^*$	$Changnivpos_{1000,-80}$
Changnivneg	$x_{i-1} > x_i \wedge x_i < x_{i+1}$	
	$\sigma_a, \sigma_b \in R_-^*$	$Changnivneg_{-500,-20}$

Le tableau II.2.1 présente les motifs utilisés pour labéliser les séries temporelles. Nous définissons 9 motifs à savoir :

— Ptpicpos : motif pour détecter un point remarquable de pic positif ;

— Ptpicneg : motif pour détecter un point remarquable de type pic négatif ;

— StartCstNeg : motif pour détecter un point remarquable de début de constant négatif ;

— StartCstPos : motif pour détecter un point remarquable de début de constant positif ;

— EndCstNeg : motif pour détecter un point remarquable de fin de constant négtif ;

— EndCstNeg : motif pour détecter un point remarquable de fin de constant positif ;

— CST : motif pour détecter un point remarquable constant ;

— Changnivpos : motif pour détecter un point remarquable de changement de niveau positif ;

— Changnivneg : motif pour détecter un point remarquable de changement de niveau négatif ;

Les seuils σ_a et σ_b sont définis par les experts. Ils sont paramétrables en fonction des séries temporelles et du domaine d'application.

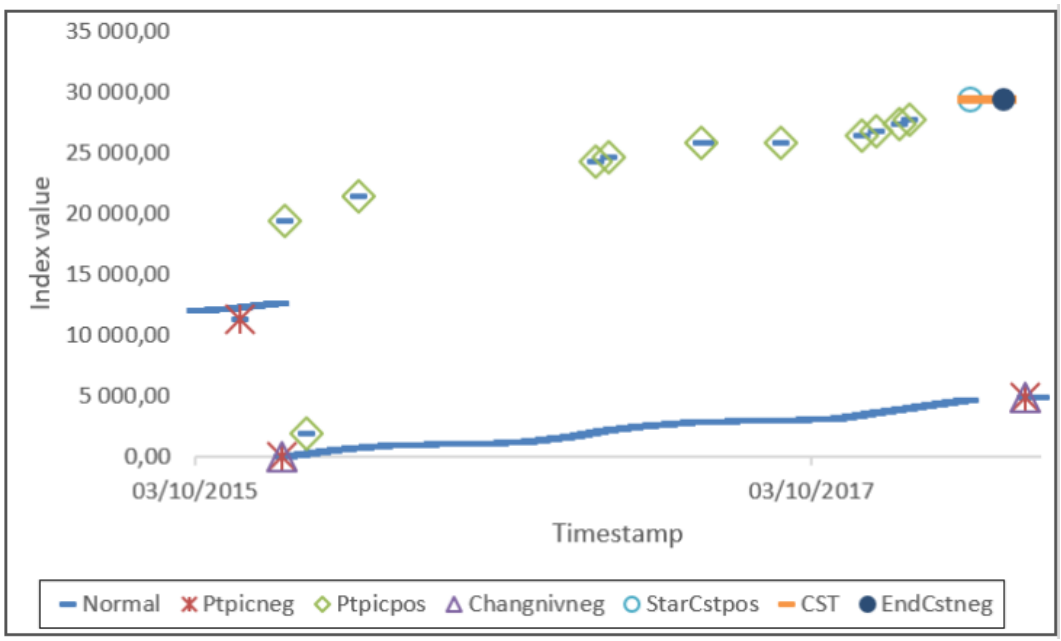


Figure II.2.2 – Labellisation d'une série de points remarquables (algorithme 1) en utilisant les motifs prédéfinis.

Exemple. La figure II.2.2 illustre un extrait d'une série temporelle labellisée des données d'index d'un compteur de calorie d'un bâtiment. Ainsi comme indiqué, elle

comporte 7 exemples de labels de différents motifs définis dans le tableau II.2.1. Un point remarquable peut être étiqueté par plusieurs motifs tels que le point représenté par la croix et le triangle sur la figure II.2.2 qui comportent 2 labels (Ptpicneg, Changnivneg).

Algorithm 1 EvaluatePattern

Input : x_{i-1}, x_i, x_{i+1}, $p = (l_p, \sigma_a, \sigma_b)$
Output : Boolean

1: **if** $p.\sigma_a > 0$ **then**
2: $leftValidated \leftarrow (v(x_i) \geq v(x_{i-1}) + p.\sigma_a \ ? \ true : false)$
3: **else if** $p.\sigma_a < 0$ **then**
4: $leftValidated \leftarrow (v(x_i) \leq v(x_{i-1}) + p.\sigma_a \ ? \ true : false)$
5: **else if** $p.\sigma_a = 0$ **then**
6: $leftValidated \leftarrow (v(x_i) = v(x_{i-1}) \ ? \ true : false)$
7: **end if**
8: **if** $p.\sigma_b > 0$ **then**
9: $rightValidated \leftarrow (v(x_i) \geq v(x_{i+1}) + p.\sigma_b \ ? \ true : false)$
10: **else if** $p.\sigma_b < 0$ **then**
11: $rightValidated \leftarrow (v(x_i) \leq v(x_{i+1}) + p.\sigma_b \ ? \ true : false)$
12: **else if** $p.\sigma_b = 0$ **then**
13: $rightValidated \leftarrow (v(x_i) = v(x_{i+1}) \ ? \ true : false)$
14: **end if**
15: $confirmedPattern \leftarrow$ *leftValidated* **and** *rightValidated*
16: **return** (confirmedPattern)

L'algorithme 1, appelé EvaluatePattern, permet d'évaluer un motif à l'aide de règles. Cette fonction prend en entrée trois points successifs notés x_{i-1}, x_i et x_{i+1} et le motif p à évaluer, et renvoie le résultat de l'évaluation indiquant si le motif est confirmé ou pas. Lors de l'évaluation, nous appliquons différentes règles de vérifications en fonction des signes de σ_a et σ_b (lignes 1–14).

L'algorithme 2 fait appel à la fonction EvaluatePattern pour traiter une série temporelle. Il prend en entrée la série temporelle initiale Ts et la liste des motifs P et renvoie une nouvelle série temporelle labellisée Ts_L. Le traitement consiste à parcourir la série temporelle et la liste des motifs. Pour chaque point x_i et pour chaque motif p_k, la fonction EvaluatePattern est appelée afin d'ajouter (ou pas) le label l du motif p_k au point évalué. A la ligne 13, le triplet (t_i, v_i, L_i) est le point remarquable formé à partir du point $x_i = (t_i, v_i)$ issu de T_S.

Algorithm 2 Détection de points remarquables

Input : $Ts = \{x_1, x_2, x_3....\}$, $P = \{p_1, p_2, p_3....\}$
Output : série temporelle labellisée Ts_L

```
1: Ts_L ← {}
2: for i in range(2..|Ts|-1) do
3:    L_i ← "Normal"
4:    for k in range(1..|P|) do
5:       if EvaluatePattern(x_{i-1},x_i,x_{i+1},p_k) then
6:          if L_i = "Normal" then
7:             L_i ← {p_k.l}
8:          else
9:             L_i ← L_i ∪ {p_k.l}
10:         end if
11:      end if
12:   end for
13:   Ts_L ← Ts_L ∪ {(t_i, v_i, L_i)}
14: end for
15: return Ts_L
```

2.3.2 Composition de motifs

Afin de détecter les anomalies, nous analysons le voisinage des points remarquables détectés lors de la phase 1. Ainsi, à partir d'un sous-ensemble de points d'une série temporelle labellisée, on construit par concaténation des labels L_i de ces points remarquables, une chaîne de labels. Sur cette chaîne nous vérifions des conditions établies sur les valeurs des points. Une anomalie est ainsi reconnue par une composition de labels et une vérification des conditions sur les points.

Définition 4.II.2 *Une anomalie* est un ou plusieurs points remarquables appartenant à un sous-ensemble de points pour lequel sont vérifiées, d'une part, une composition des labels de ces points (ordre séquentiel des labels) et, d'autre part, une condition exprimée sur les valeurs de ces points. L'anomalie est identifiée sur un ou plusieurs points de cette composition.

Pour définir une composition de labels, nous proposons une grammaire, illustrée dans la figure II.2.3, qui définit les éléments d'une composition de labels. La grammaire permet de définir les labels possibles (un ou plusieurs) sur des points successifs permettant de reconnaître une composition de labels.

La grammaire part des labels posés sur les points (<label>). Les labels peuvent être combinés sur un seul point avec des expressions logiques AND, OR et NOT (<label-comp> and <point-label>). Par exemple "l1 AND NOT l2 AND l3" désigne un point labellisé l1, non labellisé l2 et labellisé avec l3. Chaque combinaison de labels sur un

```
<composition> ::= <label-enum> ("." <label-enum>)*

<label-enum> ::= <label-comp>
     |          "(" <label-comp> ")" "?"
     |          "(" <label-comp> ")" "*"
     |          "(" <label-comp> ")" "+"

<label-comp> ::= <point-label> ("OR" <point-label>)*
     |            <point-label> ("AND" <point-label>)*
     |            <point-label>

<point-label> ::= <label>
     |            "NOT" <label>

<label> ::= list of words (remarkable points)
                  defined by patterns
```

Figure II.2.3 – Grammaire pour la définition d'une composition de labels.

point unique peut être répétée sur des points successifs par des quantificateurs : ?, + et * (<label-enum>). Par exemple "(l1) +" signifie que la composition doit comporter un ou plusieurs points successifs labellisés l1.

La composition finale de labels est créée à travers une succession d'énumération de labels séparés par "." (< composition >). Par exemple, "l1. (l2) *. (l1 OR l3)" signifie un point labellisé par l1 suivi d'aucun à plusieurs points labellisés par l2 suivi d'un point labellisé par l1 ou par l3.

Définition 5.II.2 *Une composition de labels* permettant de reconnaître une anomalie, est composée de trois parties :

— *composition* : la composition des labels de points remarquables qui est une séquence de points comportant des labels définis selon la grammaire présentée dans la figure II.2.3. Une même composition de labels peut correspondre à différentes anomalies ;

— *condition* : c'est une condition entre les valeurs des points reconnus (ceux correspondants à la séquence des labels). Cette condition est créée à l'aide des opérateurs (<>, <, <=, =, >, >=), permettant de comparer des valeurs, et des opérateurs logiques (AND/OR/NOT) permettant de combiner des comparaisons. Afin d'éviter l'utilisation de la notation $v(x_i)$, nous notons par v_i la valeur du ième point reconnu par la composition, v_1 le premier et v_n le dernier ; notons que le nombre de points impliqués dans la composition peut être variable compte tenu des quantificateurs utilisables dans la composition ;

— *conclusion* : l'anomalie identifiée pour laquelle est précisé son type (nom de l'anomalie) et la liste des valeurs (points) où se situe l'anomalie détectée.

A titre d'exemple, nous donnons des compositions de labels pour identifier des anomalies très récurrentes dans les données de capteurs : (i) anomalie de valeurs en pic positif. Cette dernière est possiblement reconnue à partir de deux compositions de labels **Label-composition 1** et **Label-composition 4** parce qu'ils ont la même partie de composition. Le but de cet exemple est de montrer l'utilité de la partie condition dans la détection d'anomalie, (ii) anomalie de valeurs en pic négatif présentée par **Label-composition 2**, et (iii) anomalie de valeurs constantes présentée par **Label-composition 3** ;

Label-composition 1

composition : Normal . Ptpicpos . Ptpicneg . Normal

condition : $v_2 > v_4$ and $v_3 > v_1$

conclusion : positive peak -> v_2

La composition "Normal . Ptpicpos . Ptpicneg . Normal" signifie qu'il existe un point labélisé "Normal" suivi d'un point "Ptpicpos" suivi de "Ptpicneg" suivi de "Normal". Si cette composition est trouvée dans la série étiquetée, nous vérifions sa condition en comparant les valeurs des points.

Dans cette composition qui est déclenchée sur 4 points successifs, il faut que la valeur du deuxième point soit supérieur au dernier point et que la valeur du troisième point soit supérieur au premier point. Si cette condition est vraie une anomalie est déclenchée.

Label-composition 2

composition : Normal . Ptpicpos . Ptpicneg . Normal

condition : $v_2 < v_4$ and $v_3 < v_1$

conclusion : negative peak -> v_3

Label-composition 2 permet de détecter une anomalie de type pic négatif.

Comme nous avons déjà indiqué, une même composition pourrait détecter différents types d'anomalies et c'est la condition qui permet d'appliquer l'une des deux. Dans cet exemple, les parties "composition" de **Label-composition 2** et **Label-composition 1** sont identiques. Cependant, la "condition" et la "conclusion" sont différentes. Ainsi, lors de l'évaluation, CoRP vérifie tout d'abord les compositions. Ensuite, il vérifie leurs conditions pour voir quelle condition est vraie pour identifier l'anomalie correspondante.

Label-composition 3

composition : Startcstpos . Cst* . Endcstpos

condition : $v_1 == v_2$ and $v_{n-1} == v_n$

conclusion : constant -> all

Dans cet exemple, "Startcstpos" signifie le début d'une constante qui est suivi de zéro

ou plusieurs constantes "Cst*", qui est suivi de fin de constante "Endcstpos". L'anomalie est l'ensemble des points de la composition.

Label-composition 4

composition : Normal . Ptpicpos . Ptpicneg AND Changnivneg . Normal

condition : $v_2 > v_4$ and $v_3 > v_1$

conclusion : positive peak -> v_2

Dans cette composition, il existe un point "Normal" suivi de "Ptpicpos" suivi d'un point qui a deux labels à la fois "Ptpicneg AND Changniv" suivi de "Normal".

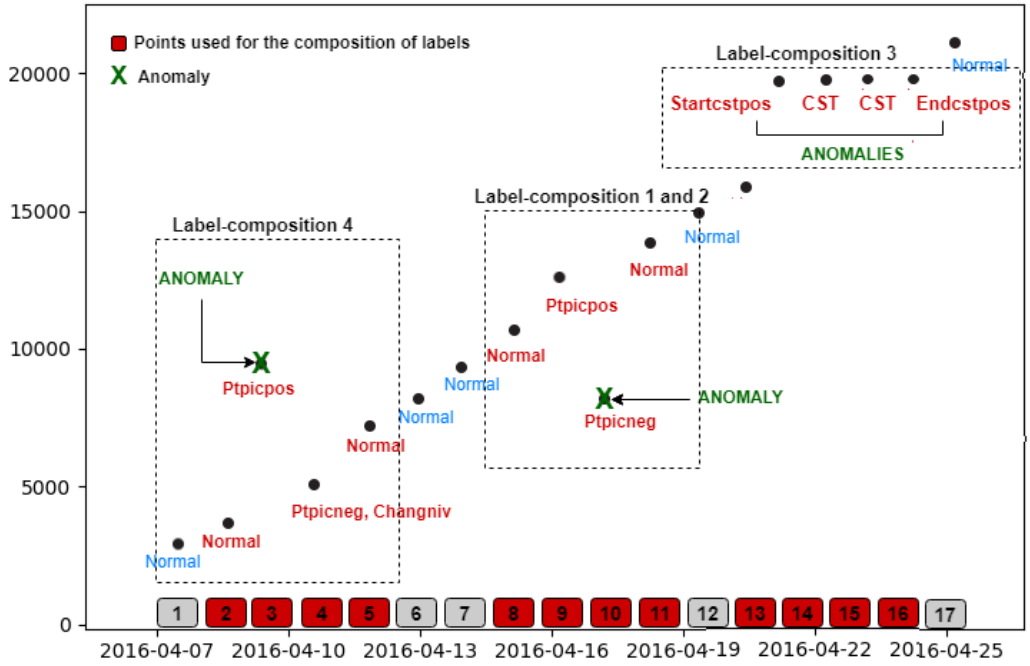


Figure II.2.4 – Résultat de la phase 2 de l'algorithme CoRP.

Exemple. Considérons les sous-ensembles de points présentés sur la figure II.2.4 en rouge. Les points d'indices 2 à 5 donnent la séquence de labels suivante : (Normal . Ptpicpos . Ptpicneg and Changniv . Normal) qui est détectée par **Label-composition 4**. Cette composition permet donc de détecter l'anomalie de pic positif en indice 3 sur la figure.

Les points d'indices 8 à 11, déclenchent **Label-composition 1** et **Label-composition 2** (la même composition avec conditions et conclusions différentes). En vérifiant **Label-composition 1**, la condition $v_9 > v_{11}$ et $v_{10} > v_8$ est fausse. Par conséquent, **Label-composition 1** n'est pas valide. Pour **Label-composition 2**, la condition $v_9 < v_{11}$ et $v_{10} < v_8$ est vraie donc la composition est valide et l'anomalie Pic Négatif est reconnue en v_{10}. Les points d'indice 12 à 16 déclenchent **Label-composition 3** permettant de

détecter un plateau constant.

2.4 Application sur les données du SGE

Nous avons mis en œuvre un algorithme capable de parcourir une liste labellisée Ts_L et vérifier, à partir de chaque point, quelles compositions de labels s'appliquent pour identifier les anomalies (et les points correspondants).

Nous avons appliqué CoRP sur le serveur de base de données du SGE. L'objectif était d'identifier les anomalies sur les données journalière et historiques des compteurs afin de faciliter les analyses des experts. En utilisant CoRP, nous avons pu localiser l'emplacement de différents types d'anomalies et préciser également le types d'anomalies comme illustré dans la figure II.2.5.

Chrono	Name	Value	Quality	TS	Anomalie	Type_Anomalie
131896116000000000	CVC.INS...	2745,301	192	2018-12-18 13:00:00.000	0	Normal
131896152000000000	CVC.INS...	2745,407	192	2018-12-18 14:00:00.000	0	Normal
131896188000000000	CVC.INS...	2745,4951	192	2018-12-18 15:00:00.000	1	constant
131896198785250000	CVC.INS...	2745,4951	24	2018-12-18 15:17:59.000	1	constant
131896296000000000	CVC.INS...	2745,7939	192	2018-12-18 18:00:00.000	0	Normal
131896332000000000	CVC.INS...	2745,875	192	2018-12-18 19:00:00.000	0	Normal
131896368000000000	CVC.INS...	2745,9199	192	2018-12-18 20:00:00.000	0	Normal
131896404000000000	CVC.INS...	185159	86	2018-12-18 21:00:00.000	1	anomalie point positif
131896440000000000	CVC.INS...	2745,988	192	2018-12-18 22:00:00.000	0	Normal
131896476000000000	CVC.INS...	2746,0181	192	2018-12-18 23:00:00.000	0	Normal
131896512000000000	CVC.INS...	2746,05	192	2018-12-19 00:00:00.000	0	Normal
131896548000000000	CVC.INS...	2746,0869	192	2018-12-19 01:00:00.000	0	Normal
131896584000000000	CVC.INS...	2746,1201	192	2018-12-19 02:00:00.000	0	Normal
131896620000000000	CVC.INS...	2746,1531	192	2018-12-19 03:00:00.000	0	Normal
131896656000000000	CVC.INS...	2746,1941	192	2018-12-19 04:00:00.000	0	Normal
131896692000000000	CVC.INS...	155920,09	86	2018-12-19 05:00:00.000	1	anomalie point positif
131896728000000000	CVC.INS...	2746,332	192	2018-12-19 06:00:00.000	0	Normal
131896764000000000	CVC.INS...	2746,5601	192	2018-12-19 07:00:00.000	0	Normal

Figure II.2.5 – Application de CoRP sur les données du SGE.

2.5 Synthèse de la première contribution : CoRP

Ce chapitre présente notre contribution pour « la détection d'anomalies » dans les séries temporelles uni-variées par la modélisation de motifs et de compositions à l'aide des experts. L'approche vise à identifier différents types de valeurs aberrantes observées lors de déploiements réels. L'objectif ciblé est de maximiser le nombre d'anomalies détectées et de minimiser les fausses alertes et les faux positifs.

Ainsi, notre proposition, CoRP, repose sur la détection d'anomalies basée sur les motifs. Elle consiste à labéliser tous les points remarquables qui présentent un comportement inhabituel à l'aide de motifs. Ensuite, par compositions de labels, elle identifie précisément les multiples anomalies présentes dans les séries temporelles univariées. Cette approche nécessite l'expertise du domaine pour pouvoir définir efficacement les motifs. En effet, la détermination des seuils σ_a et σ_b exigent une exploration des données afin de comprendre le comportement des points remarquables qui peuvent y exister. Également, la composition de labels est issue des connaissances et retours des experts qui ont tendance à analyser le voisinage des points remarquables pour détecter les anomalies. Contrairement aux méthodes de la littérature présentées dans le tableau I.2.3, CoRP demande une expertise métier pour être appliquée. Par conséquent, notre approche peut ne pas être complètement adaptée aux attentes des usagers qui préféreraient une approche plus automatique. Cependant, elle répond à leur besoin principal en étant précise dans la détection de multiples anomalies. De plus, une piste d'amélioration pourrait s'appuyer sur l'apprentissage automatique des motifs pour apprendre les seuils ou encore apprendre les compositions de labels.

Pour conclure notre approche offre plusieurs avantages. Elle offre la possibilité de détecter différents types d'anomalies observées lors de déploiements réels. L'approche permet aussi d'identifier précisément la localisation des points aberrants. Elle donne une information sur le type d'anomalie trouvé dans les séries temporelles uni-variées et elle est fiable pour la détection d'anomalies multiples.

2.6 Conclusion

Notre méthode CoRP est composée de deux étapes : elle marque (labels) tous les points remarquables présents dans la série temporelle sur la base de motifs de détection, puis, elle identifie précisément les multiples anomalies présentes à partir de compositions de labels. Cette approche nécessite l'expertise du domaine d'application pour pouvoir définir efficacement les motifs et les compositions de labels. Bien qu'elle demande une bonne expertise métier pour être appliquée, CoRP a l'avantage d'être précise pour détecter différents types d'anomalies, localiser les points où se trouve l'anomalie et générer peu d'erreurs comme nous allons le montrons dans la partie III lors des évaluations. Notre travail a été publié dans trois articles : une conférence internationale ICEIS'2019 (« best student paper award ») (Ben Kraiem *et al.*, 2019), une conférence nationale Inforsid'2019 (Ben Kraiem *et al.*, 2019) et une contribution à un ouvrage de synthèse (Ben Kraiem *et al.*, 2019).

Dans la partie suivante, nous allons présenter notre deuxième contribution qui permet d'automatiser le processus d'étiquetage et de générer automatiquement des règles de classification.

3 CDT : Composition-based Decision Tree

« La meilleure façon de prédire l'avenir, c'est de le créer. »

Peter Ferdinand Drucker (1909 — 2005)

Table des matières

3.1 Introduction

Les règles interprétables par l'homme pour la détection d'anomalies se réfèrent à des données anormales présentées dans un format (règles) qui est intelligible pour les analystes. L'apprentissage de ces règles est une tâche difficile, et seuls quelques travaux abordent la question des différents types d'anomalies dans les séries chronologiques. Dans le chapitre précédent, nous avons proposé une méthode qui permet de résoudre l'enjeu de la détection de multiples anomalies en utilisant des motifs pour labéliser les séries temporelles et des compositions de motifs pour identifier les anomalies.

Dans ce chapitre, nous cherchons à résoudre un autre enjeu qui est la production automatique de règles pour la détection d'anomalies afin d'aider les experts à prendre une décision. Nous conservons l'idée principale des motifs et des compositions. Notre objectif est de rendre la labélisation automatique et d'apprendre les compositions de motifs afin de générer des règles.
Pour ce faire, nous décrivons notre méthode appelée CDT basée sur des motifs qui permet de générer automatiquement un ensemble réduit de règles compréhensibles par l'homme. Ainsi, nous proposons une labélisation automatique des séries temporelles à travers des motifs. Puis nous construisons un arbre de décision basé sur des compositions de motifs. De plus nous utilisons une optimisation bayésienne pour éviter le réglage manuel des hyper-paramètres et nous définissons une mesure de qualité pour évaluer à la fois l'exactitude et l'intelligibilité des règles produites.

3.2 Contexte et motivation

Dans un contexte réel, les experts analysent les anomalies qu'ils détectent dans les courbes et construisent manuellement des règles de décision pour détecter les futures occurrences de ces anomalies. Cependant, comme la quantité de données collectées augmente, les règles de décision deviennent plus complexes à définir, ce qui rend l'analyse plus difficile.

L'extraction automatique de règles et la détection en temps opportun de ces différentes valeurs aberrantes peuvent être d'un intérêt considérable pour un expert. Pour surmonter ce défi, des algorithmes d'apprentissage de règles ont été proposés (Barakat et Diederich, 2005; Singh et Gupta, 2014). Le déploiement de tels systèmes pourrait révéler des informations compréhensibles aux utilisateurs, afin d'expliquer la cause première des anomalies, mieux que les algorithmes de type boîte noire.

Un autre défi important réside dans la présence de différents types d'anomalies qui

peuvent se produire dans les séries chronologiques. Ces différences peuvent dépendre du domaine d'application (Chandola *et al.*, 2009; Aggarwal, 2015). Ainsi, un problème difficile dans les approches de détection d'anomalies est de prendre en compte la diversité de toutes les anomalies existantes comme nous l'avons expliqué dans le chapitre I.2.

Pour relever ces défis, nous proposons une méthode d'apprentissage automatique pour générer des règles interprétables par l'homme pour la détection d'anomalies dans les séries temporelles uni-variées, appelée Composition-based Decision Tree (CDT). Cette méthode utilise des séquences de motifs (modèles) pour identifier des points remarquables correspondant à de multiples anomalies. Les compositions (séquences) de motifs existant dans des séries temporelles sont apprises grâce à un arbre de décision généré en interne, puis simplifiées à l'aide d'une algèbre booléenne pour produire des règles intelligibles. Nous utilisons l'optimisation bayésienne des hyper-paramètres (deux hyper-paramètres) pour obtenir les meilleurs hyper-paramètres pour notre méthode. L'approche vise à trouver le meilleur compromis entre une haute précision pour les détections d'anomalies et un ensemble minimisé de règles plus facilement interprétables par l'homme.

3.3 Méthodologie CDT

Dans cette section, nous décrivons notre méthode Composition-based Decision Tree (CDT) pour l'extraction de règles et la détection d'anomalies. Tout d'abord, les séries temporelles sont prétraitées en effectuant une normalisation des valeurs et possiblement des ré-échantillonnages. Ensuite, nous créons une formalisation de 9 variations anormales correspondant aux différents types d'anomalies sous forme de motifs. Les motifs prédéfinis sont utilisés pour créer automatiquement des séries chronologiques étiquetées. Par la suite, un arbre de décision modifié est créé pour construire un classificateur de détection d'anomalies et pour générer des règles de décision. Nous simplifions les règles produites et nous évaluons la qualité des règles et, enfin, nous ajustons les hyper-paramètres en utilisant une optimisation bayésienne. Nous décrivons chaque étape, illustrée sur la figure II.3.1, plus en détail dans les sections suivantes.

3.3.1 Prétraitement des séries chronologiques

Nous prétraitons les séries temporelles uni-variées. Les séries chronologiques sont collectées à partir de différents capteurs et les valeurs des mesures sont sur des plages

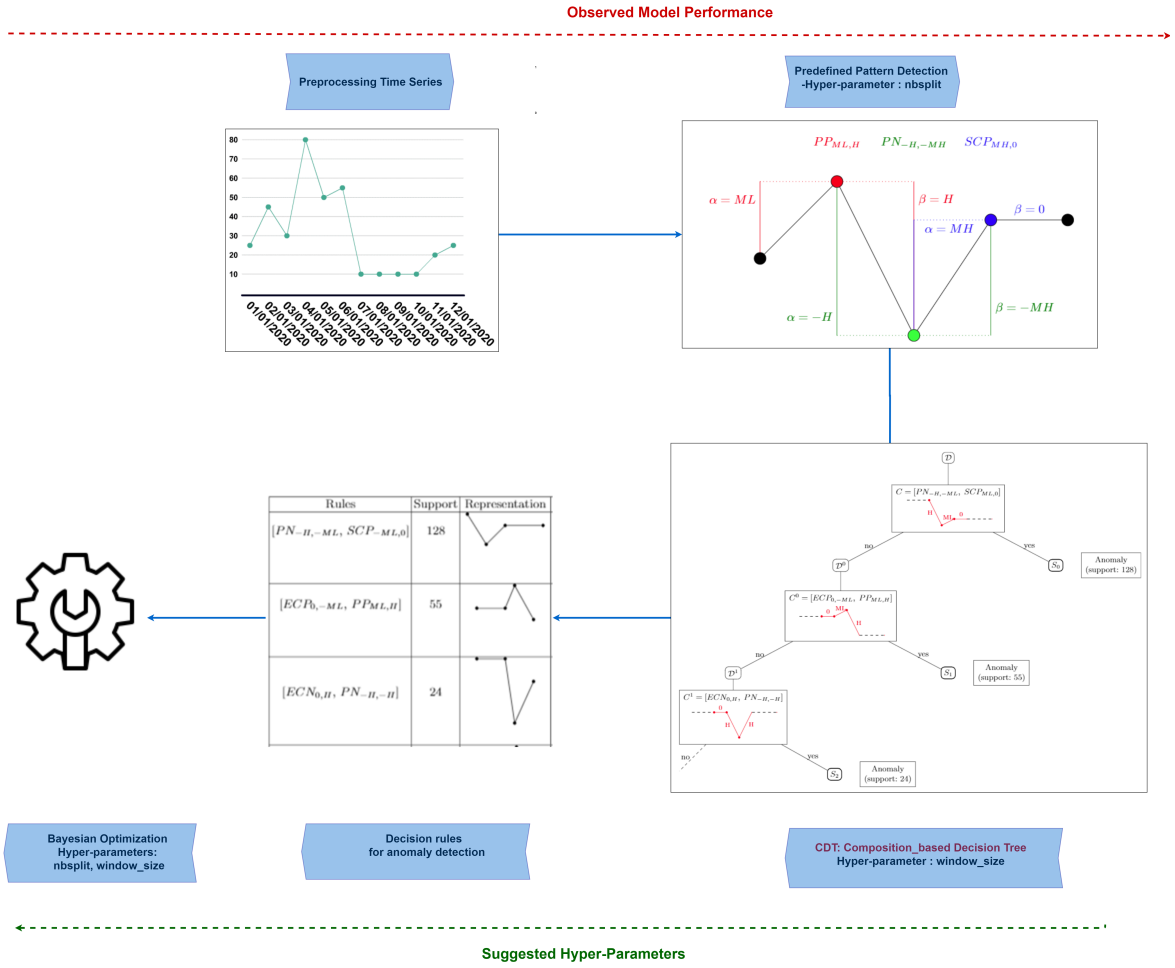


Figure II.3.1 – Les différentes étapes de l'approche CDT.

différentes. Pour obtenir une invariance d'échelle et de décalage, et pour assurer une meilleure stabilité du modèle, nous normalisons chaque série temporelle continue Ts à des valeurs comprises entre 0 et 1. Les données ainsi obtenues ont été normalisées à l'aide de l'équation suivante :

$$norm(x_i) = \frac{x_i - min(Ts)}{max(Ts) - min(Ts)} \tag{3.1}$$

tel que x_i est une mesure (valeur de point) d'une série temporelle Ts.

Dans l'analyse et l'exploration des séries temporelles, le ré-échantillonnage peut également être utilisé. Le sous-échantillonnage (« downsampling » en anglais) est le processus de réduction de la fréquence d'échantillonnage des données. En effet, il consiste à fabriquer une série comportant moins d'échantillons qu'un signal d'origine. Par exemple, admettons qu'un capteur de température envoie des données à un système de supervision toutes les secondes, l'expert peut avoir besoin des données suivant une

fréquence d'une heure ou d'une semaine au lieu de secondes. Ceci lui facilite les analyses en limitant le nombre de points x_i. À l'aide du sous-échantillonnage, plusieurs points de données dans une plage de temps pour une seule série temporelle sont agrégés à l'aide d'une fonction mathématique, par exemple la moyenne, en une seule valeur à un horodatage aligné.

3.3.2 Étiquetage des séries chronologiques

Dans cette section, nous cherchons à utiliser les variations typiques entre les points successifs pour formaliser des motifs que nous allons utiliser pour étiqueter les points remarquables dans les séries temporelles. La labélisation avec les motifs permettre de générer par la suite des règles intelligibles et plus lisibles par les experts.

Considérons trois points successifs d'une série temporelle Ts notés x_{i-1}, x_i, x_{i+1}. Trois points successifs permettent de définir 9 variations possibles comme listées dans le tableau II.3.1 à savoir, PP (Pic positif), PN (Pic négatif), SCP (Start Constant Positive), SCN (Start Constant Negative), ECP (End Constant Positive), ECN (End Constant Negative), CST (Constant), VP (Variation Positive) et VN (Variation Negative).
Soit une série temporelle Ts normalisée (valeurs entre 0 et 1), nous supposons que chacune de ces variations peut avoir des magnitudes différentes entre [-1,1].

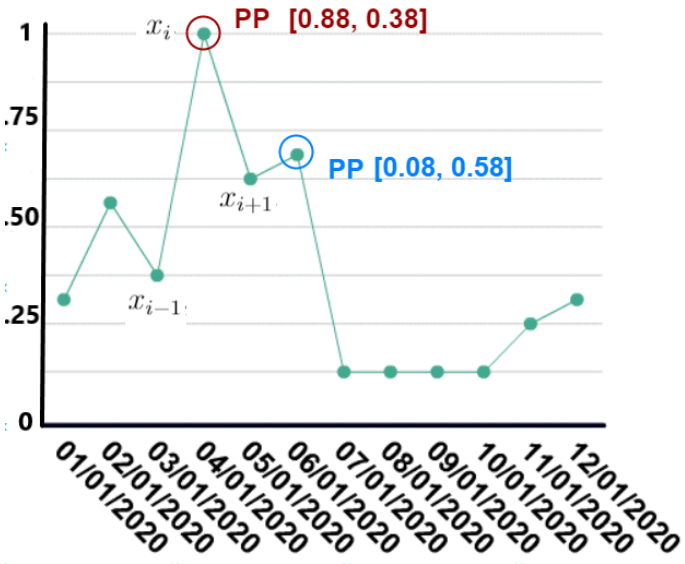


Figure II.3.2 – Exemple de variation de type Pic Positif (PP). Il y a deux modèles marqués $PP_{[0.88,0.38]}$ et $PP_{[0.08,0.58]}$ correspondant à multiple pics positifs mais avec des amplitudes différentes.

Afin d'affiner l'efficacité de détection de notre approche, on peut distinguer pour

chaque variation plusieurs intervalles en [-1,1]. Par exemple, la figure II.3.2 illustre une variation PP avec différentes magnitudes ([0.88, 0.38] et [0.08, 0.58]. Les valeurs [0.88, 0.38] et [0.08, 0.58] sont la variation du point x_i par rapport x_{i-1} et x_i par rapport x_{i-1}. Nous remarquons que la magnitude de la première variation est plus grande que la deuxième variation.

Hyper-paramètre (δ). On note δ l'hyper-paramètre, utilisé pour distinguer les différentes grandeurs des 9 variations (PP, PN, SCP, SCN, ECP ECN, CP, VN et CST).

Pour un δ donné, nous construisons des intervalles $2\delta + 1$ (en prenant leurs limites entre [-1, 1]) décrivant l'ampleur de variation des points : δ intervalles pour les variations positives (dans l'intervalle $]0, 1]$), δ intervalles pour les variations négatives (dans l'intervalle $[-1, 0]$), et un cas particulier pour l'absence de variation (égal à 0).

Cet hyper-paramètre permet d'avoir des motifs d'amplitudes plus ou moins fines pour capturer tout changement de points dans les séries. δ sera déterminé automatiquement en utilisant une optimisation bayésienne.

Définition 1.II.3 On définit un *motif* noté $P = (l, \alpha, \beta)$ où l, est un nom (ou une étiquette) identifiant le motif, et α et β sont deux intervalles possibles de [-1,1].
Pour chaque points successifs x_{i-1}, x_i, x_{i+1}, le point x_i est labélisé par un motif uniquement si $x_i - x_{i-1} \in \alpha \wedge x_i - x_{i+1} \in \beta$. Dans ce cas, x_i est étiqueté avec l.

Par exemple, avec $\delta = 2$, nous construisons 5 intervalles : $\delta^{]0,0.5]}$, $\delta^{]0,5,1]}$, $\delta^{[-0,5,0[}$, $\delta^{[-1,-0.50[}$ et $\delta^{[0,0]}$. Par souci de simplicité, dans le reste du chapitre, nous considérons uniquement la notation avec $\delta = 2$ (les autres valeurs de δ ne résulteront qu'en une plus grande variété d'intervalles et de motifs), et nous désignons chaque intervalle comme suit :

— $\delta^{]0,0.5]}$: Low (L =]0,0.5]),

— $\delta^{]0.5,1]}$: High (H =]0.5,1]),

— $\delta^{[-0.5,0[}$: -Low (-L = [-0.5,0[),

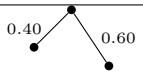
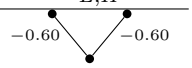
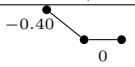
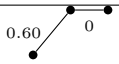
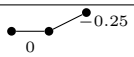
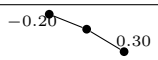
— $\delta^{[-1,-0.50[}$: -High (-H = [-1,-0.5[),

— $\delta^{[0,0]}$: cas spécial, Zero (Z = 0).

En utilisant ces 5 intervalles (pour $\delta = 2$) et les 9 variations résumées dans le tableau II.3.1 nous obtenons 23 motifs possibles comme suit :

— pour la variation PP, nous pouvons avoir 4 motifs : $PP_{L,H}$, $PP_{H,H}$, $PP_{H,L}$, $PP_{L,L}$. Le même principe est suivi pour la variation PN ;

Tableau II.3.1 – Les types de variations pour la labélisation.

Variation	Définition	Motif	Exemple
PP	$x_{i-1} < x_i \wedge x_i > x_{i+1}$		
	$\alpha, \beta \in \{L, H\}$	$PP_{\alpha,\beta}$	$PP_{L,H}$
PN	$x_{i-1} > x_i \wedge x_i < x_{i+1}$		
	$\alpha, \beta \in \{-L, -H\}$	$PN_{\alpha,\beta}$	$PN_{-H,-H}$
SCN	$x_{i-1} > x_i \wedge x_i = x_{i+1}$		
	$\beta \in \{Z\}$ $\alpha \in \{-L, -H\}$	$SCN_{\alpha,\beta}$	$SCN_{-L,0}$
SCP	$x_{i-1} < x_i \wedge x_i = x_{i+1}$		
	$\beta \in \{Z\}$ $\alpha \in \{L, H\}$	$SCP_{\alpha,\beta}$	$SCP_{H,0}$
ECN	$x_{i-1} = x_i \wedge x_i > x_{i+1}$		
	$\alpha \in \{Z\}$ $\beta \in \{L, H\}$	$ECN_{\alpha,\beta}$	$ECN_{0,L}$
ECP	$x_{i-1} = x_i \wedge x_i < x_{i+1}$		
	$\alpha \in \{Z\}$ $\beta \in \{-L, -H\}$	$ECP_{\alpha,\beta}$	$ECP_{0,-L}$
CST	$x_{i-1} = x_i \wedge x_i = x_{i+1}$		
	$\alpha, \beta \in \{Z\}$	$CST_{\alpha,\beta}$	$CST_{0,0}$
VP	$x_{i-1} < x_i \wedge x_i < x_{i+1}$		
	$\alpha \in \{L, H\}$ $\beta \in \{-L, -H\}$	$VP_{\alpha,\beta}$	$VP_{L,-L}$
VN	$x_{i-1} > x_i \wedge x_i > x_{i+1}$		
	$\alpha \in \{-L, -H\}$ $\beta \in \{L, H\}$	$VN_{\alpha,\beta}$	$VN_{-L,L}$

— pour la variation VP, on obtient 3 motifs seulement : $VP_{L,H}$, $VP_{H,L}$, $VP_{L,L}$. Le motif $VP_{H,-H}$ ne peut pas être appliqué. En effet, $VP_{H,-H}$ correspond à $VP_{]0.5,1],[-1,-0.5[}$. Étant donné que les valeurs 0.5 et -0.5 sont exclues dans les intervalles H et -H, n'importe quelle combinaison de valeurs de ces intervalles va dépasser la plage de valeurs réelles de la série temporelle qui est entre 0 et 1.
Le même principe est suivi pour la variation VN.

— pour la variation SCN, nous avons 3 motifs possibles : $SCN_{-L,0}$ et $SCN_{-H,0}$. Le même principe est suivi pour SCP, ECN et ECN ;

— pour la variation CST nous avons un seul motif $CST_{0,0}$.

En utilisant ces intervalles, nous pouvons créer par exemple un motif $PP_{L,H}$ où PP est un pic positif avec $\alpha =]0, 0.5]$ (marqué L) et $\beta =]0.5, 1]$ (marqué H). Inversement, nous pourrions définir un motif $PN_{-L,-H}$ dans le cas de pic négatif.

Exemple La figure II.3.3 illustre des points remarquables représentés par diffèrents motifs nommés $PP_{L,H}$, $PN_{-H,-H}$, $SCN_{H,0}$.

— $PP_{L,H}$ est un pic positif tel que : PP présente la variation $\alpha = L$, $\alpha =]0, 0.5]$ (marqué L) et $\beta =]0.5, 1]$ (marqué H).

— $PN_{-H,-H}$ est un pic positif tel que : $\alpha = -H$ et $\beta = -H$;

— $SCP_{H,0}$ est un début de constant positif tel que : $\alpha = H$ et $\beta = 0$.

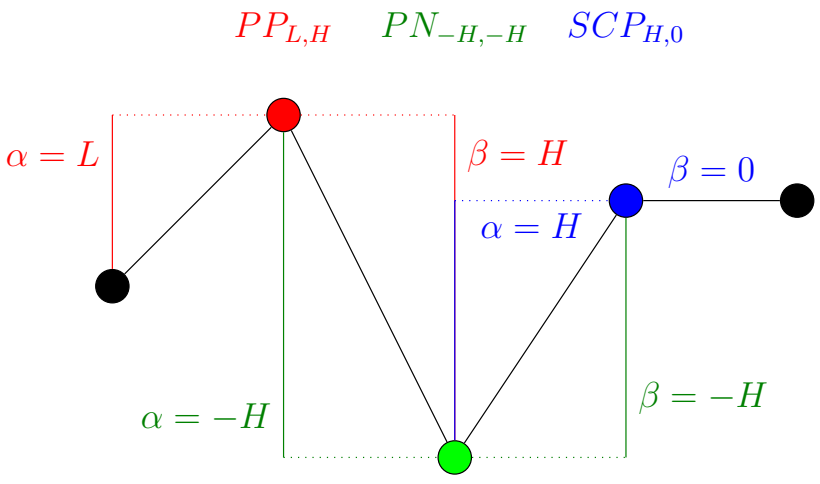


Figure II.3.3 – Exemple de différents motifs.

La figure II.3.4 représente une série temporelle des données de consommation du compteur de calories d'un bâtiment. Elle montre des exemples de différentes grandeurs de motif telles que $PP_{L,H}$, $PP_{L,L}$ et $PP_{H,H}$. En utilisant ces modèles, nous pouvons automatiquement étiqueter chaque point remarquable de la série.

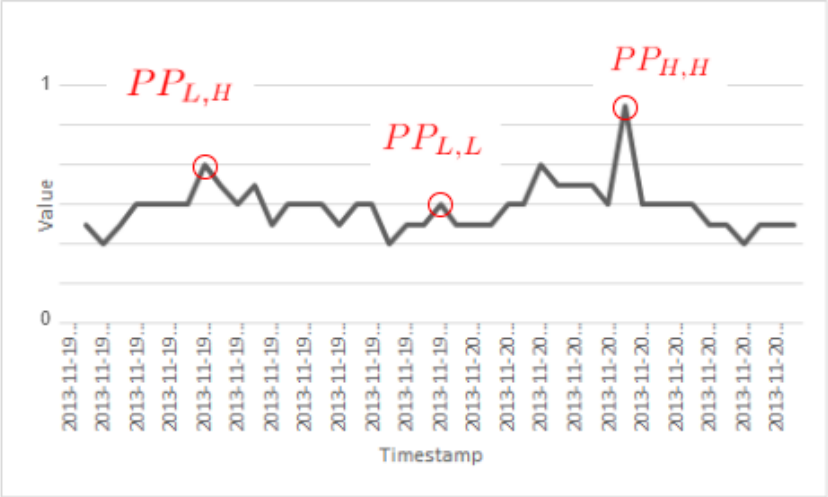


Figure II.3.4 – Exemple de motif. Il y a trois motifs marqués $PP_{L,H}$, $PP_{L,L}$ et $PP_{H,H}$ correspondant à des pics positifs mais avec des amplitudes différentes.

Définition 2.II.3 Une *série temporelle étiquetée* notée $Tsb = \{l_1, ..., l_N\}$ avec $N = n-2$ où chaque point x_i d'une série temporelle initiale (Ts) est remplacé par l'étiquette de son motif correspondant.

Notons que pour ce chapitre nous adaptons des notations avec L,H,-H,-L dans le cas général où $\delta = 2$, les étiquettes sont notées avec les intervalles explicites. Par exemple, si $\delta = 4$, nous aurons pour PP les intervalles suivants]0,0.25], [0.25,0.50],]0,50,0.75],]0.75,1].

3.3.3 Composition-based Decision Tree

3.3.3.1 Problématique

Bien que les arbres de décision soient conçus pour produire des règles compréhensibles, ce n'est pas toujours le cas, et ils ne sont pas totalement adaptés à nos besoins :

— Les règles peuvent devenir inintelligibles si les arbres sont grands. Le grand nombre de règles ainsi obtenu rend les règles difficilement interprétables. De plus, certaines règles peuvent ne pas avoir de sens pour l'utilisateur.

— L'arbre de décision considère les variables (features) sans aucun ordre lors de la division du jeu de données. En revanche, les experts pensent toujours en fonction de points successifs. Ainsi, nous adaptons le principe des arbres de décision pour prendre en compte l'ordre entre les variables. Nous cherchons à construire des règles basées sur des compositions de variables ordonnées similaires à la grammaire proposée dans la première contribution, mais de manière automatisée.

Pour résoudre les problèmes mentionnés ci-dessus, nous proposons une version adaptée des arbres de décision, pour générer des règles interprétables pour la détection des anomalies.

3.3.3.2 Les arbres de décision

Les arbres de décision (AD) sont utilisée dans l'exploration de données comme un outil d'aide à la décision. Ils emploient une représentation hiérarchique de la structure des données sous forme des séquences de décisions (tests) en vue de la prédiction d'une variable ou d'une classe.

Cet algorithme d'apprentissage supervisé est bien connu pour sa simplicité et sa compréhensibilité. Il est induit à partir des observations d'apprentissage composées de valeurs de variables ou caractéristiques et d'une étiquette de classe (variable cible). Un arbre est construit en divisant les données d'apprentissage en sous-ensembles en choisissant la variable qui partitionne le mieux les données d'apprentissage selon un critère d'évaluation. Ce critère caractérise l'homogénéité des sous-ensembles obtenus par division de l'ensemble de données. Parmi ces critères, nous pouvons citer l'entropie de Shannon et l'indice de diversité de Gini. Ce processus est répété de manière récursive sur chaque sous-ensemble dérivé jusqu'à ce que toutes (ou presque) les instances d'un sous-ensemble soient dans la même classe (Su et Zhang, 2006).

Ce principe de construction est appelé "Top-Down", c'est à dire que l'arbre est construit de la racine vers les feuilles suivant une approche algorithmique gloutonne et récursive. Dans un arbre de décision chaque nœud interne (ou nœud de décision) décrit un test sur une variable d'apprentissage, chaque branche (ou arc) représente un résultat du test, et chaque feuille contient la valeur de la variable cible (une étiquette de classe).

L'arbre de décision définit un classifieur qui se traduit en terme de règles de décision, mutuellement exclusives et ordonnées (sous forme de si-alors-sinon).

3.3.3.3 Description de CDT

Dans un arbre classique, chaque nœud réalise un test portant sur une variable dont le résultat indique la branche à suivre dans l'arbre. Au contraire dans notre arbre, chaque noeud effectue un test portant sur une composition de motifs (séquence ordonnée de points remarquables) représentant ainsi une suite de variables ordonnées.

Dans cette section, nous décrivons notre méthode CDT. La construction de l'arbre étend la construction de l'arbre de décision classique tels que CART (Breiman *et al.*,

1984), C4.5 (Quinlan , 1993).
Nous nous orientons vers une classification sous forme d'arbres binaires permettant de classifier 2 catégories de classes (normale et anomalie). L'entrée de notre arbre de décision est sous forme de fenêtres glissantes (voir chapitre I.1 pour plus de détails) de taille fixe créées à partir de la série chronologique étiquetée.

Définition 3.II.3 Un *ensemble d'observations* noté $D = \{d_1, d_2, ..., d_{N-\omega+1}\}$ représente le résultat de la coupe de Tsb par fenêtre glissante de taille ω, et un pas fixe de 1. Ainsi les différentes observations sont :

$D = \{\{l_1, ..., l_\omega\}, \{l_2, ..., l_{\omega+1}\}, ..., \{l_{N-\omega+1}, ..., l_N\}\}$.
Soit M le nombre de classes auxquelles les observations sont associées. Dans notre contexte, nous considérons deux classes (M = 2) : la classe anormale (observation avec anomalie), ou la classe normale (observation sans anomalie). Chaque observation $d_i \in D$ est associée à une seule classe annotée $class(d_i)$.

Hyper-paramètre (ω). On note $\omega <= N/2$ une taille de fenêtre tel que N représente la taille de la série temporelle labélisée Tsb. Cet hyper-paramètre sera déterminé automatiquement à l'aide de l'optimisation bayésienne.

Afin de déterminer la probabilité de distribution des observations sur les classes d'un noeud, plusieurs fonctions d'hétérogénéité, ou d'impureté peuvent être définies telles que l'indice de Gini ou l'entropie. Nous optons dans notre méthode pour l'indice de Gini (Singh et Gupta, 2014) comme mesure d'impureté d'un sous ensemble d'observations $D_j \subseteq D$.

Définition 4.II.3 L'*indice d'impureté de Gini*, noté $G(D_j)$, fournit une mesure de la qualité de l'ensemble d'observations D_j selon la distribution des observations dans les classes. La métrique d'impureté est minimale (égale à 0) quand toutes les observations appartiennent à une même classe, et elle est maximale (égale à 0.5) si il y a autant d'éléments de chaque classe.

L'indice d'impureté de Gini est défini comme :

$$G(D_j) = \sum_{k=1}^{M} p_k(1 - p_k), \tag{3.2}$$

où p_k est la fraction d'observations appartenant à la classe k tel que $p_k = |d_i \in D_j / class(d_i) = k| / |D_j|$.

À partir d'une observation, nous définissons une composition utilisée comme caractéristique pour diviser un nœud en deux sous-nœuds.

Définition 5.II.3 Une *composition* notée c est une séquence d'étiquettes existantes

dans une observation d_i. En utilisant le symbole $\subseteq_o$, nous notons $c \subseteq_o d_i$.

Exemple. Considérons $d = \{l_1, l_2, l_3, l_4, l_5, l_6\}$.

— $c = \{l_2, l_3, l_4\} \subseteq_o d$,

— $c = \{l_3, l_2, l_4\} \not\subseteq_o d$,

— $c = \{l_1, l_2, l_3, l_4, l_5, l_6\} \subseteq_o d$.

Nous introduisons également des notations supplémentaires : $c \in_o D$ lorsque $\forall d \in D, c \subseteq_o d$, et $c \notin_o D$ lorsque $\forall d \in D, c \not\subseteq_o d$.

Un arbre de décision est construit sur la base des variables qui ont le gain d'information le plus élevé. Lors de la création du CDT, les compositions sont comparées en fonction du gain d'information qu'elles procurent.

Définition 6.II.3 Un *Gain d'information* noté IG, permet de mesurer la qualité de partition d'un sous-ensemble de compositions et la quantité d '"informations" qu'une composition nous donne sur la classe.

$$IG(D_j, c) = G(D_j) - (\frac{|D_{inc}|}{|D_j|}G(D_{inc}) + \frac{|D_{exc}|}{|D_j|}G(D_{exc})) \tag{3.3}$$

où $|D_{inc}|$, $|D_{exc}|$ and $|D_j|$ sont respectivement la taille de D_{inc}, D_{exc} et D_j. D_{inc} et D_{exc} sont deux sous-ensembles de D_j où $D_{inc} = \{d \in D_j | c \subseteq_o d\}$ et $D_{exc} = \{d \in D_j | c \not\subseteq_o d\}$.

Le processus de CDT est décrit par l'algorithme 3. Pour construire un arbre de décision, on définit un nœud de l'arbre comme un quadruplet (comme indiqué à la ligne 1 de l'algorithme 3) :

— *observations* : l'ensemble des observations considérées dans ce nœud ;

— *composition* : utilisée pour diviser *observations* en nœuds-fils ;

— *childTrue* : le nœud d'observations satisfaisant la *composition* ;

— *childFalse* : le nœud d'observations qui ne satisfont pas la *composition*.

Nous introduisons une fonction $list\_of\_all\_possible\_compositions()$ pour calculer toutes les compositions déduites de D_j (ligne 6 dans l'algorithme 3). Pour chaque composition, nous calculons le gain d'information pour diviser un nœud (ligne 7-15). A la ligne 16, si $G(D_j) \neq 0$ signifie que l'ensemble des observations du nœud est impure (les observations sont de classes différentes). De plus, $maxGain = 0$ signifie qu'une composition qui divise l'ensemble des observations a déjà été trouvée : dans ce cas, nous créons un nœud N_{inc} qui représente la branche positive du nœud ($c \subseteq_o D_j$), et N_{exc} qui représente la branche négative ($c \not\subseteq_o D_j$) (ligne 16–25). Nous répétons ces étapes jusqu'à ce qu'il n'y ait plus de nœuds à traiter (ligne 3–26).

Algorithm 3 CDT : Composition-based Decision Tree

Input : $D = \{d_1, d_2, ..., d_{N-\omega+1}\}$ a set of observations
Output : N_{root} the root node of CDT

$N_{root} \leftarrow Node(D, null, null, null)$
$q \leftarrow [N_{root}]$ *// construct the queue of nodes to split*
while $q \neq \emptyset$ **do**
  $N_j \leftarrow q.pop()$ *// dequeue the first node from the queue*
  $D_j \leftarrow N_j.observations$
  $C_j \leftarrow list\_of\_all\_possible\_compositions(D_j)$
  $maxGain \leftarrow 0$
  $c_{best} \leftarrow null$
  // Choose the composition that has the best Gain
  for all $c \in C_j$ **do**
    if $IG(D_j, c) > maxGain$ **then**
      $maxGain \leftarrow IG(D_j, c)$
      $c_{best} \leftarrow c$
    end if
  end for
  if $G(D_j) \neq 0$ and $maxGain \neq 0$ **then**
    $D_{inc} \leftarrow \{d \in D_j | c_{best} \subseteq_o d\}$
    $D_{exc} \leftarrow \{d \in D_j | c_{best} \not\subseteq_o d\}$
    $N_{inc} \leftarrow Node(D_{inc}, null, null, null)$
    $N_{exc} \leftarrow Node(D_{exc}, null, null, null)$
    $q.append(N_{inc})$ *// enqueue child nodes*
    $q.append(N_{exc})$ *// enqueue child nodes*
    $N_j.composition \leftarrow c_{best}$
    $N_j.childTrue \leftarrow N_{inc}$
    $N_j.childFalse \leftarrow N_{exc}$
  end if
end while
return N_{root}

Exemple. La figure II.3.5 illustre un exemple de résultat de CDT. Cet arbre est un extrait des données réelles du SGE. Le nœud racine nommé $\mathcal{D}_1$, représente l'ensemble des observations d'apprentissage utilisées pour la construction de l'arbre. Les feuilles représentent des étiquettes de classe et les branches représentent des conjonctions de compositions, qui mènent à ces étiquettes de classe. Dans cette figure II.3.5, l'arbre est composé de 3 partitions (splits) construisant un ensemble de 3 feuilles $\mathcal{S} = \{S_1, S_2, S_3\}$.

3.3.3.4 Génération de règles pour la détection d'anomalies

Cette étape consiste à produire un système de règles de classification à partir de l'arbre construit via l'ensemble des chemins partant de la racine de l'arbre et arrivant

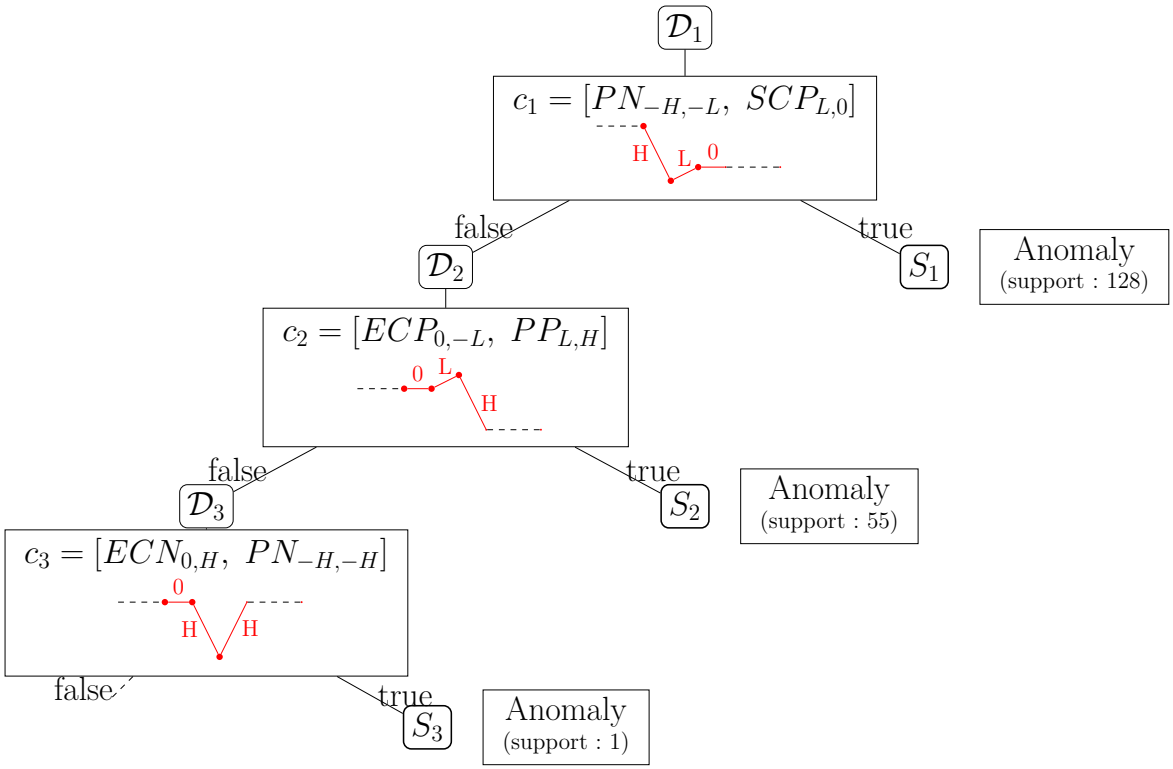


Figure II.3.5 – Illustration d'un arbre obtenu avec CDT.

à chacune des feuilles. Ainsi, nous convertissons l'arbre issu de CDT en un ensemble de règles de décision. Nous considérons uniquement les « feuilles pures » conduisant à la classe d'anomalies.

Définition 7.II.3 Un *prédicat de règle*, noté Rs est une branche de l'arbre de décision où un chemin menant à la classe d'anomalies. C'est une conjonction ("ET" logique) de tests rencontrés. Il est construit en combinant (conjonction) les compositions successives c_i ou $\neg c_i$ de feuilles au nœud racine. Pour chaque branche positive ($c_i \in_o D_j$), la composition positive c_i est déduite alors qu'une composition négative $\neg c_i$ est déduite d'une branche négative ($c_i \notin_o D_j$).

Exemple. Trois prédicats de règles sont produits à partir de l'arbre de la figure II.3.5.

- $R_{S_1} : c_1 = [PN_{-H,-L},\ SCP_{L,0}]$
- $R_{S_2} : c_2 \wedge \neg c_1 = [ECP_{0,-L},\ PP_{L,H}] \wedge \neg[PN_{-H,-L},\ SCP_{L,0}]$
- $R_{S_3} : c_3 \wedge \neg c_2 \wedge \neg c_1 = [ECN_{0,H},\ PN_{-H,-H}] \wedge \neg[ECP_{0,-L},\ PP_{L,H}] \wedge \neg[PN_{-H,-L},\ SCP_{L,0}]$.

Nous utilisons des notations abusives, $c_i \wedge \neg c_j$ qui signifie que pour une observation d sur une série temporelle, on vérifie $c_i \subseteq_o d \wedge c_j \not\subseteq_o d$. Donc, bien que abusivement noté sous forme d'expressions logiques pour des raisons de simplicité, un prédicat de

règle définit une liste de compositions qu'une observation doit contenir ou pas pour être considérée comme anomalie.

Définition 8.II.3 Une règle, notée $\mathcal{R}$, est une disjonction de prédicats de règles. Par exemple, comme indiqué dans la figure II.3.5 : $\mathcal{R} = R_{S_1} \vee R_{S_2} \vee R_{S_3} = (c_1) \vee (c_2 \wedge \neg c_1) \vee (c_3 \wedge \neg c_2 \wedge \neg c_1)$.

3.3.4 Simplification des règles

L'objectif d'avoir des prédicats de règles lisibles par l'expert nous amène à considérer leur longueur (nombre de compositions, nombre d'étiquettes). Les simplifications envisagées visent à réduire le nombre de compositions composant les prédicats de règles. Nous cherchons à éliminer les compositions qui sont redondantes ou ne semblent pas performantes pour prédire la classe. Cet élagage permet d'éliminer certains tests de la partie conditionnelle d'un prédicat de règle ou éliminer un prédicat de règle entier.

Une façon de minimiser les prédicats de règles de CDT est de post-traiter la règle produite par simplification de l'expression logique à l'aide des règles de l'algèbre booléenne. L'origine de l'algèbre de Boole est du mathématicien britannique George Boole (Taylor, 1954). Il existe trois opérateurs de base :

— Non/Not, noté $\neg a$ ou $\overline{a}$, qui inverse la valeur de la variable a ;

— Et/And, noté $a \cdot b$ ou $a \wedge b$ qui retourne 1 si a et b sont égales à 1, sinon retourne 0 ;

— Ou/Or, noté a + b ou $a \vee b$ qui retourne 1 si a ou b est à 1, sinon retourne 0

A partir des propriétés de l'algèbre de Boole, nous pouvons transformer la fonction logique pour la simplifier (Taylor, 1954). Parmi ces propriétés nous avons utilisé la commutativité présentée par l'équation suivante :

$$a \cdot b = b \cdot a \quad et \quad a + b = b + a \tag{3.4}$$

Également, il existe plusieurs lois de l'algèbre de Boole tels que le Théorème d'allégement, théorème d'absorption, Théorème de Morgan etc. Nous avons appliqué la loi d'allégement pour simplifier nos règles en utilisant la règle suivante :

$$a + \overline{a} \cdot b = a + b \quad et \quad (a + b) \cdot \overline{a} = b \cdot \overline{a} \tag{3.5}$$

L'expression logique peut être écrite sous deux formes : la forme " somme de produits " et la forme " produit de sommes ". Les règles générées par CDT sont sous la forme

de «somme des produits» de fonctions booléennes. En effet, il s'agit d'une disjonction de compositions.

Ainsi en utilisant les équations 3.5 et 3.4, nous pouvons simplifier la règle $\mathcal{R}$ générée à partir de l'arbre de la figure II.3.5 comme suivant :

$$\begin{aligned}\mathcal{R} &= R_{S_1} \vee R_{S_2} \vee R_{S_3} \\ &= (c_1) \vee (c_2 \wedge \neg c_1) \vee (c_3 \wedge \neg c_2 \wedge \neg c_1) \\ &= (c_1) \vee (\neg c_1 \wedge c_2) \vee (\neg c_1 \wedge \neg c_2 \wedge c_3) \\ &= (c_1) \vee (c_2) \vee (c_3)\end{aligned}$$

Exemple. Prenons l'exemple des trois prédicats de règles générés à partir de la figure II.3.5.

$$\begin{aligned}\mathcal{R} &= R_{S_1} \vee R_{S_2} \vee R_{S_3} \\ &= [PN_{-H,-L}, SCP_{L,0}] \vee ([ECP_{0,-L}, PP_{L,H}] \wedge \neg[PN_{-H,-L}, SCP_{L,0}]) \vee \\ &([ECN_{0,H}, PN_{-H,-H}] \wedge \neg[ECP_{0,-L}, PP_{L,H}] \wedge \neg[PN_{-H,-L}, SCP_{L,0}]) \\ &= [PN_{-H,-L}, SCP_{L,0}] \vee (\neg[PN_{-H,-L}, SCP_{L,0}] \wedge [ECP_{0,-L}, PP_{L,H}]) \vee \\ &(\neg[PN_{-H,-L}, SCP_{L,0}] \wedge \neg[ECP_{0,-L}, PP_{L,H}] \wedge [ECN_{0,H}, PN_{-H,-H}]) \\ &= [PN_{-H,-L}, SCP_{L,0}] \vee [ECP_{0,-L}, PP_{L,H}] \vee [ECN_{0,H}, PN_{-H,-H}]\end{aligned}$$

Dans notre approche, cette simplification interbranche est appliquée jusqu'à ce qu'il n'y ait plus de simplification à faire. Cela nous permet de minimiser le nombre de compositions dans une règle.

3.3.5 Mesure de qualité

Dans (Barakat et Diederich, 2005), les auteurs définissent des critères de qualité de règles afin d'évaluer les règles extraites de l'algorithme SVM (Support vector machines) tels que la compréhensibilité et la précision des règles. Ainsi, les auteurs considèrent un ensemble de règles comme précis s'il peut classer correctement de nouveaux exemples. La compréhensibilité d'un ensemble de règles est déterminée en mesurant la taille de l'ensemble de règles (en termes de nombre de règles) et le nombre d'antécédents par règle.

Dans (Daud et Corne, 2009), les auteurs ont étudié les performances des algorithmes de classification de la littérature tels que des algorithmes d'induction de règles

et d'arbres de décision. Pour leurs résultats expérimentaux, ils s'intéressent au pourcentage d'instances correctement classées des algorithmes (pourcentage de précision) et à la lisibilité qui est calculée en fonction du nombre de règles ou à la taille des arbres produits par les classificateurs (nombre de noeud).

Dans notre travail, nous visons à générer des règles à la fois précises et compréhensibles. Nous supposons qu'une règle compréhensible doit être courte et doit contenir un nombre réduit de différentes étiquettes. Pour cette raison, nous avons défini les critères suivants :

— $\mathcal{I}(c)$ pour caractériser la *qualité d'une composition* c en fonction de sa longueur et du nombre de motifs utilisés ;

— $\mathcal{M}(\mathcal{I}_{R_S})$ pour caractériser la *qualité de prédicat de règle* R_S en fonction du nombre de compositions de R_S et de la qualité de chacune des compositions $\mathcal{I}(c)$;

— $\mathcal{Q}(\mathcal{R})$ pour caractériser la *qualité d'une règle* $\mathcal{R}$ en fonction de la qualité des prédicat de règles et de leur support.

Nous calculons l'interprétabilité d'une composition en utilisant l'équation suivante :

$$\mathcal{I}(c) = 1 - \frac{L_c.N_L}{\omega.MaxL} \tag{3.6}$$

où $L_c = |c|$ désigne la longueur d'une composition c (nombre d'étiquettes qui composent c), N_L est le nombre d'étiquettes uniques utilisées dans une composition, ω est la taille maximale de la fenêtre et $MaxL$ est le nombre maximal d'étiquettes.
Ensuite, nous calculons l'interprétabilité moyenne d'un prédicat de règles (conjonction de compositions) en utilisant l'équation suivante :

$$\mathcal{M}(\mathcal{I}_{R_S}) = \frac{1}{Nc} \sum_{k=1}^{Nc} \mathcal{I}(c_k) \tag{3.7}$$

où Nc est le nombre de compositions dans le prédicat de règle R_S.
La qualité des règles extraites est calculée comme suivant :

$$\mathcal{Q}(\mathcal{R}) = \frac{1}{S} \sum_{i=1}^{nb} S_{R_{S_i}}.\mathcal{M}(\mathcal{I}_{R_{S_i}}) \tag{3.8}$$

où nb est le nombre de prédicats de règle dans $\mathcal{R}$, $S_{R_{S_i}}$ est le support du prédicat de règle R_{S_i} (vrai positif) et S est le support de tous les prédicats de règles (vrai positif et vrai négatif). Étant donné qu'un prédicat de règles qui a un support élevé est plus important, nous multiplions l'interprétabilité moyenne d'un prédicat de règles avec son support.

3.3.6 Sélection automatique des hyper-paramètres

Comme mentionné dans les sections précédentes, CDT a deux hyper-paramètres qui sont le nombre de divisions (δ) pour spécifier les grandeurs des motifs et la longueur de la fenêtre (ω) indiquant la taille des observations de CDT.

Le réglage manuel nécessite une connaissance préalable et prend du temps, car il s'agit d'une recherche par force brute (Snoek *et al.*, 2012). En effet, la recherche manuelle teste des ensembles d'hyper-paramètres définis par l'utilisateur, qui doit utiliser ses connaissances pour identifier les paramètres qui amélioreront le résultat souhaité. Pour surmonter ce problème, des algorithmes de recherche automatique ont été proposés dans la littérature comme la recherche par grille ou la recherche aléatoire (Wu *et al.*, 2019). La recherche par grille est une recherche exhaustive car elle calcule toutes les combinaisons de valeurs possibles pour les hyper-paramètres. Bien qu'il puisse donner de bons résultats, son coût reste élevé.
La recherche aléatoire essaie des combinaisons aléatoires de valeurs. Elle est plus efficace que la recherche de grille, mais peut ne pas trouver l'ensemble optimal d'hyper-paramètres.
Dans la recherche par grille et la recherche aléatoire, nous essayons les configurations de manière aléatoire et aveugle. Le prochain essai est indépendant de tous les essais effectués auparavant. En revanche, le réglage automatique des hyper-paramètres permet de connaître la relation entre les valeurs d'hyper-paramètres et les performances du modèle afin de faire un choix plus judicieux pour les valeurs d'hyper-paramètres suivants. Le but est de minimiser le nombre d'essais tout en trouvant un bon optimum.

Dans cette optique, nous utilisons l'optimisation bayésienne (Wu *et al.*, 2019; Xia *et al.*, 2017) pour trouver les meilleurs hyper-paramètres pour notre modèle. L'objectif de l'optimisation des hyperparamètres dans l'apprentissage automatique est de trouver les hyperparamètres d'un algorithme d'apprentissage automatique donné, qui renvoient les meilleures performances mesurées sur un ensemble de validation. En effet, nous cherchons à trouver une configuration (c'est-à-dire un ensemble de paramètres) qui maximise une métrique de performance ou une fonction objectif.

L'optimisation bayésienne est une approche probabiliste qui, contrairement à la recherche aléatoire ou par grille, gardent une trace des résultats d'évaluation passés, qu'elle utilise pour former un modèle probabiliste sur la fonction objectif. Ce modèle est appelé fonction de substitution ou «surrogate » en anglais. La fonction de substitution est la représentation de probabilité de la fonction objectif construite à l'aide des évaluations précédentes. Cette fonction est beaucoup plus facile à optimiser que la fonction objectif puisque le temps passé à sélectionner les hyper-paramètres est moins

important par rapport au temps passé dans la fonction objectif (Shahriari *et al.*, 2015).

Le principe est de trouver le prochain ensemble d'hyper-paramètres à évaluer sur la fonction objectif réelle, en sélectionnant les hyper-paramètres qui fonctionnent le mieux sur la fonction de substitution. Ainsi, l'optimisation des hyper-paramètres comporte les étapes suivantes :

1. Définir un domaine de recherche pour les hyper-paramètres ;
2. Définir une fonction objectif qui prend en entrée les hyper-paramètres et produit un score que nous voulons maximiser (ou minimiser) ;
3. Construire un modèle de probabilité de substitution de la fonction objectif ;
4. Trouvez les hyper-paramètres qui fonctionnent le mieux sur le substitut. Un critère, appelé fonction de sélection ou acquisition, pour évaluer les hyper-paramètres à choisir ensuite dans le modèle de substitution ;
5. Appliquer ces hyper-paramètres à la fonction objectif ;
6. Mettre à jour le modèle de substitution en intégrant les nouveaux résultats.

Il existe plusieurs choix pour le modèle de substitution tels que Gaussian Processes, Random Forest Regressions, et Tree Parzen Estimators (TPE) ainsi que pour la fonction de sélection des hyper-paramètres, tels que Expected Improvement et Upper Confidence Bound (Snoek *et al.*, 2012; Shahriari *et al.*, 2015).

L'optimisation des hyper-paramètres se présente sous forme de l'équation ci-dessous (Snoek *et al.*, 2012) :

$$h^* = \arg\max_{h \in H} F(h) \tag{3.9}$$

où $F(h)$ représente une fonction objectif à maximiser, $h*$ est l'ensemble des hyper-paramètres (δ, ω) à optimiser et h peut prendre n'importe quelle valeur dans l'espace de recherche H.

Pour optimiser le compromis entre les performances de détection et la bonne qualité des règles, nous avons défini la fonction objectif $F(h)$ comme la F-mesure (performance de détection) pondérée par la mesure de qualité des règles $\mathcal{Q}(\mathcal{R})$.

$$F(h) = F1(h).\mathcal{Q}(\mathcal{R}) \tag{3.10}$$

où $F1(h)$ est la F-mesure (la moyenne harmonique de la précision et du rappel) de la performance de classification obtenue avec l'ensemble de paramètres (h).

Dans notre travail, nous utilisons pour calculer la fonction de substitution, les processus gaussiens. Ils nous permettent de générer pour chaque point une distribution

de probabilité caractérisée par une moyenne (la valeur la plus probable) et un écart-type (la mesure de la dispersion probable de la valeur autour de la moyenne). Pour la fonction de sélection, nous utilisons Upper Confidence Bound. Cette fonction associe à chaque point de l'espace de recherche un potentiel pour être l'optimal. Trouver le nouveau point à évaluer implique donc d'évaluer notre fonction de sélection pour tout notre espace de recherche. Cependant ces évaluations seront beaucoup moins coûteuses en comparaison à l'évaluation de la fonction objectif.

3.4 Synthèse de la deuxième contribution : CDT

Ce chapitre présente une méthode d'apprentissage automatique appelée Composition-based Decision Tree (CDT) pour la détection d'anomalies. Elle génère des règles interprétables par l'homme basées sur une formalisation de 9 variations permettant de définir des motifs pour étiqueter les séries temporelles automatiquement et détecter les points remarquables. Cet étiquetage augmente la lisibilité des règles. Compte tenu de cet étiquetage de séries temporelles qui augmente la lisibilité des règles, un arbre de décision est ensuite construit, en considérant les nœuds comme des compositions de motifs avec le gain d'information le plus élevé.
L'entrée du CDT est construite en créant des fenêtres coulissantes de taille fixe, où les hyperparamètres de division et de longueur de fenêtre sont automatiquement calculés via l'optimisation bayésienne.
L'arbre est ensuite converti en un ensemble de règles de décision, pour lesquelles une mesure de qualité est définie. Ceci est basé sur l'interprétabilité de la composition, qui tient compte de la longueur de la règle et de son nombre d'étiquettes.

Pour conclure notre approche induit plusieurs avantages. Elle permet de générer des règles interprétables et simples avec de bonnes performances de détection d'anomalies. Avec une telle sortie, les experts peuvent analyser et expliquer les règles de détection pour prendre une décision ou même ajuster les règles (par exemple, combiner des règles, généraliser des règles, simplifier ou compléter des règles) en fonction de leurs connaissances de terrain.

3.5 Conclusion

Notre méthode CDT proposée est une méthode d'apprentissage automatique qui génère des règles compréhensibles par les experts pour la détection d'anomalies multiples dans les séries temporelles uni-variées. L'approche est basée sur un arbre de

décision modifié. En utilisant l'optimisation bayésienne, nous avons optimisé les hyper-paramètres de manière à maximiser à la fois la qualité des règles et les performances de classification. Notre travail est publié dans deux conférences internationales RCIS'2020 (Ben Kraiem *et al.*, 2020) et [EDBT'2021] (Ben Kraiem *et al.*, 2021).

Troisième partie

Implantation et expérimentation des propositions

1 Introduction

« Ce qui est affirmé sans preuve peut être nié sans preuve. »

Euclide de Mégare (v. 450 av. J.-C. — v. 380 av. J.-C.)

Table des matières

Cette troisième partie du mémoire rend compte de la démarche de validation expérimentale mise en place pour asseoir les contributions présentées dans la partie II. Nous présentons d'abord un aperçu des expérimentations réalisées puis nous décrivons les données utilisées pour l'évaluation de chaque proposition

1.1 Aperçu des expérimentations réalisées

Notre démarche consiste à valider expérimentalement les deux contributions proposées CoRP et CDT.

— **CoRP.** Afin d'évaluer la performance de l'algorithme CoRP pour la détection d'anomalies, une expérimentation a été conduite à partir des données du monde réel (SGE dataset) et des données de benchmark issues de la littérature (ARIMA dataset). En se comparant avec des algorithmes de la littérature, notre approche se montre plus robuste et plus précise pour détecter tous les types d'anomalies observées dans des déploiements réels ;

— **CDT.** Pour évaluer notre solution, notre algorithme est comparé à des méthodes concurrentes de détection d'anomalies sur des jeux de données réels (SGE datasets) et des benchmarks (Yahoo datasets). Notre méthode réalise de meilleures performances montrant la robustesse de l'approche. Les résultats montrent un bon équilibre entre la performance de l'arbre en terme de détection d'anomalies (précision) et de production de règles intelligibles (lisibilité).

1.2 Description des datasets

1.2.1 SGE datasets

Le domaine d'application traité dans ce mémoire est le réseau de capteurs du Service de gestion et d'exploitation (SGE) du campus de Rangueil rattaché au rectorat de Toulouse. Ce service exploite et entretient le réseau de distribution à partir des données liées aux différentes installations. Plus de 600 capteurs de différents types de fluides (calories, eau, air comprimé, électricité et gaz), disséminés dans plusieurs bâtiments, sont gérés par les systèmes de supervision du SGE. Dans nos expérimentations nous nous sommes concentrés sur les données de calories et d'électricité.

Les mesures de ces capteurs sont rassemblées à une fréquence régulière et représentent les *index* (lectures de capteurs). Ces derniers sont ensuite utilisés pour mesurer

les *quantités d'énergie consommées* (par différences de valeurs d'index successives). Nous avons pu identifier les types d'anomalies et les points concernés (points remarquables) présents dans les données de capteurs de calorie grâce aux connaissances acquises auprès des experts du SGE et à travers une inspection manuelle d'un ensemble de capteurs de même type que les capteurs étudiés.

Les mesures de calories collectées chaque jour pendant plus de trois ans par 25 capteurs déployés dans différents bâtiments soit environ 33536 observations au total. Ces mesures contiennent 586 anomalies de différents types tels que des pics positifs (PP), des pics négatifs (PN), des variations soudaines (VN, VP) et des constantes (CST). Ces anomalies représentent 1.75% des données. Les défauts présentés dans la figure I.2.4 sont extraits de ces mêmes ensembles de données.
Les mesures d'électricité sont collectées toutes les heures depuis 10 ans (96074 observations au total). Elles présentent une consommation électrique, d'un compteur, dans un bâtiment. Différents types d'anomalies existent, par exemple, des constantes (CST) ou des pics (PP, PN). Il y a au total 10343 anomalies dans le jeu de données sur l'électricité soit 10.77% des données.

L'anomalie prédominante dans ces données est constituée par les valeurs constantes suite à un arrêt de capteurs. Nous avons également trouvé parmi ces valeurs plusieurs constantes avec un décalage. Généralement, une constante avec un décalage de niveau commence par un pic positif ou négatif. Ensuite, il existe beaucoup de changements anormaux tels que des pics positifs ou négatifs. Enfin, il existe des changements de niveau dus au changement de capteur.

Nous avons utilisé les données d'index et de consommation de calorie pour évaluer CoRP quant à CDT, nous avons utilisé les données de consommation de Calorie et d'électricité.

1.2.2 ARIMA datasets

Afin d'évaluer l'algorithme CoRP dans un autre contexte, nous avons utilisé les ensembles de données proposés dans le package d'implémentation de la méthode ARIMA (Tsay , 1988). Parmi ces données, nous avons exploré :

— les données de HIPC (Harmonised Indices of Consumer Prices). Ces ensembles de données représentent les indices harmonisés des prix à la consommation dans la zone euro.

— les données IPI (Industrial Production Indices). Ces données représentent les indices de la production industrielle dans le secteur manufacturier des pays de

l'Union monétaire européenne (Tsay , 1988).

Chacun de ces ensembles de données contient plusieurs séries temporelles qui présentent des données mensuelles de 1995 à 2013. Chacune de ces séries contient 229 mesures avec 5 anomalies en HIPC comme illustré dans la figure III.1.1, et 4 anomalies en IPI. Ces anomalies sont variées : AO (Additive Outlier), TC (Temporary Changes) ou LS (Level Shift). Nous avons rapporté ces anomalies par rapport à notre typologie d'anomalies dans le tableau I.2.1. Ainsi, AO correspond à des pics, TC correspond au bruit et LS correspond à un changement de niveau.

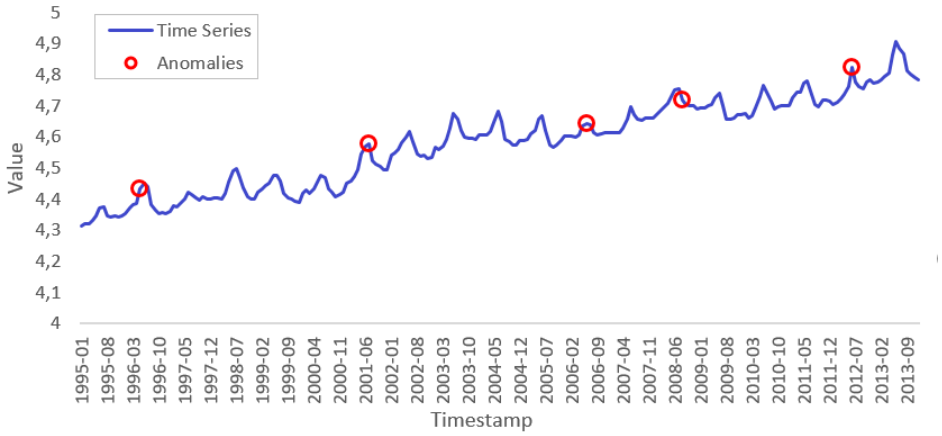


Figure III.1.1 – Exemple de DataSets HIPC avec des anomalies de types AO et TC.

1.2.3 Yahoo's S5 Webscope Dataset

Yahoo a créé un programme « Yahoo Webscope » qui est une bibliothèque de référence d'ensembles de données intéressants pour une utilisation non commerciale par des universitaires et d'autres scientifiques. L'ensemble de données Webscope S5, qui est accessible au public sur (Laptevand et Amizadeh, 2015), se compose de 371 fichiers répartis en quatre catégories, nommées A1 / A2 / A3 et A4, chacune contenant respectivement 67/ 100 / 100/ 100 fichiers. A1 Benchmark est basé sur le trafic de production réel des services Web réels, tandis que les classes A2, A3 et A4 contiennent des données d'anomalies synthétiques. Ces ensembles de données sont représentés par des séries chronologiques en unité d'une heure. Les informations sur les anomalies de vérité terrain sont disponibles pour toutes les séries chronologiques.

Les valeurs anormales dans A1 Benchmark ont été étiquetées manuellement et les données présentent une variation de trafic relativement importante par rapport aux autres ensembles de données disponibles dans les autres catégories (A2, A3, A4). Il y a total 94778 valeurs de trafic dans 67 fichiers différents dont 1669 sont anormales

(soit 1.76% des données). Les anomalies dans les jeux de données synthétiques sont insérées à des positions aléatoires. A2 Benchmark contient 142002 observations avec 466 anomalies (soit 0.33% des données) tandis que 168000 valeurs existent dans les Benchmarks A3 et A4 avec respectivement 943 et 837 anomalies (soit 0.56% et 0.20% des données respectivement).

Dans ce mémoire, nous avons utilisé les données de toutes les catégories (A1, A2, A3 et A4) pour évaluer notre algorithme CDT. Les caractéristiques de tous les ensembles de données sont décrites dans le tableau III.3.1.

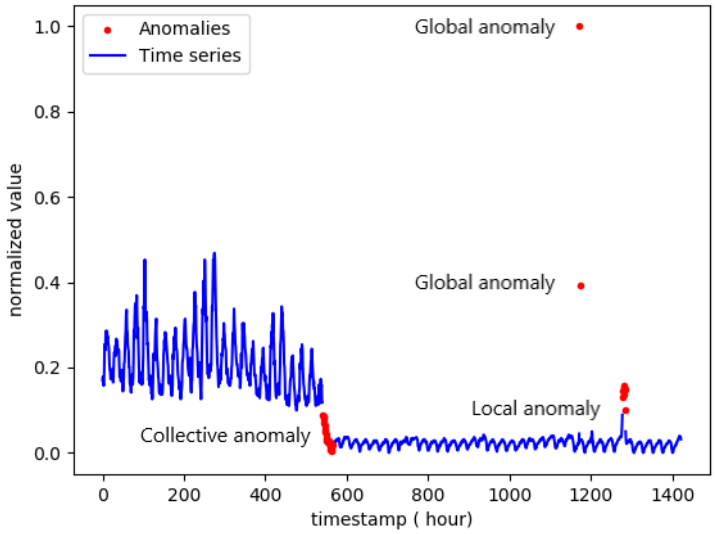


Figure III.1.2 – Exemple de DataSets Yahoo : des séries temporelles du trafic Web.

La figure III.1.2 présente un exemple d'anomalies, de vérité terrain, observées dans les ensembles de données de trafic Web. Comme illustré dans cette figure, il existe différents types d'anomalies : (i) anomalie globale dans laquelle l'anomalie apparaît à l'extérieur du trafic et présente un pic, (ii) anomalie locale lorsque l'anomalie existe à l'intérieur du trafic, (iii) anomalie collective qui présente irrégularités à long terme. Nous avons fait la correspondance de ces types d'anomalies par rapport aux anomalies que nous recherchons dans le tableau I.2.1. Ainsi, l'anomalie globale correspond à des pics. L'anomalie locale et collective correspond à un bruit et un changement dans les variations.

2 Expérimentation de la méthode basée sur les motifs CoRP

"What I hear, I forget. What I see, I remember. What I do, I understand."
Confucius (551 av. J.-C. — 479 av. J.-C.)

Table des matières

2.1 Introduction

Après avoir décrit la « méthode CoRP » dans la partie II, nous l'expérimentons dans le présent chapitre en compétition avec différentes méthodes de détection d'anomalies. Ces expérimentations ont été menées sur les données réelles du SGE les ensembles de données de référence de ARIMA présentés dans le chapitre III.1.

2.2 Méthodologie de l'expérimentation

Nous explicitions dans cette section les méthodes que nous avons utilisées lors de notre expérimentation, le protocole expérimental, ainsi que les résultats obtenus sur les données réelles de notre étude de cas et des données issues de la littérature.

2.2.1 Exploration des méthodes de détection existantes

Dans notre étude, nous avons exploré cinq méthodes appartenant à quatre techniques différentes pour détecter les types d'anomalies observées dans notre application.

— Méthode basée sur les règles : nous avons utilisé deux règles pour détecter les anomalies courtes (changement anormal) et les anomalies constantes (pas de variation) Sharma *et al.* (2010). *La règle d'anomalie courte* traite la série temporelle en comparant à chaque fois deux observations successives : on détecte une anomalie si la différence entre ces observations est supérieure à un seuil donné. Pour déterminer automatiquement le seuil de détection, nous avons utilisé l'approche basée sur l'histogramme Ramanathan *et al.* (2006). *La règle d'anomalie constante* calcule l'écart-type pour un ensemble d'observations successives. Si cette valeur est égale à zéro l'ensemble est déclaré comme anomalie.

— Méthode basée sur la densité : cette approche consiste à comparer la densité autour d'un point par rapport à la densité de ses voisins locaux. Breunig *et al.* (2000) ont proposé l'algorithme LOF. Dans cette méthode, les scores des anomalies sont mesurés en utilisant un facteur de valeur aberrante locale, qui est le rapport entre la densité locale autour de ce point et la densité locale autour de ses plus proches voisins. Le point dont la valeur LOF est élevée, est déclaré comme anomalie.

— Méthode basée sur les statistiques : premièrement, nous avons utilisé la méthode SH-ESD, qui utilise la décomposition de séries temporelles STL (décomposition saisonnière et tendance utilisant Loess) développée par Cleveland *et al.* (1990) pour diviser le signal de série chronologique en trois parties : saisonnier, tendance et

résidu. Des techniques de détection d'anomalies résiduelles sont ensuite appliquées telles que l'algorithme ESD en utilisant des métriques statistiques. Deuxièmement, nous avons utilisé la méthode Change Point pour détecter le changement de niveau.

— Méthode basée sur l'analyse des séries temporelles : le principe de cette approche est d'utiliser les corrélations temporelles pour modéliser et prédire les valeurs de la série temporelle. Nous avons utilisé le modèle ARIMA (AutoRegressive Intergrated Moving Average) pour la création du modèle de prédiction selon l'approche décrite par Chen et Liu (1993). Une mesure de capteur est comparée à sa valeur prédite pour déterminer si elle est une anomalie.

Il existe des implémentations open source pour des algorithmes (cela a guidé nos choix pour ne pas refaire les algorithmes) tels que LOF, ARIMA, S-H-ESD et Change Point ((Hochenbaum *et al.*, 2017), (López-de-Lacalle, 2016), (Rosner, 1983), (Aminikhanghahi et Hori, 2017)) que nous avons utilisés pour les expérimentations. En revanche, nous avons mis en œuvre d'autres approches (règle courte et règle constante) en fonction des sources disponibles.

Le tableau III.2.1 représente la synthèse des méthodes que nous avons explorées pour détecter chaque type d'anomalies présentées dans le tableau I.2.1. Comme nous l'avons indiqué dans le chapitre I.2, les algorithmes ne peuvent pas détecter différents types d'anomalies. Pour cette raison, nous les avons évalué selon les catégories d'anomalies qu'ils peuvent détecter.

Tableau III.2.1 – Les méthodes de détection d'anomalies étudiées.

Type d'anomalies	**Méthodes de détection**
Pics	Règle courte, ARIMA , LOF , S-H-ESD
Bruit	ARIMA , LOF , S-H-ESD
Plateau	Règle constante
Changement de niveau	ARIMA, Change Point

2.2.2 Protocole expérimental

La figure III.2.1 illustre la démarche que nous avons suivie pour appliquer notre algorithme CoRP sur les données du SGE (les index et consommation des données de calorie). Ainsi, notre méthodologie d'évaluation est composée de deux étapes. La première étape consiste à explorer et analyser, avec l'aide des experts, les points remarquables et leurs voisinages dans un exemple de série temporelles. La deuxième étape consiste à appliquer CoRP, sur les données de test, pour étiqueter les points

remarquables (PR) en utilisant les motifs, puis identifier les anomalies à partir des compositions. En fonction des points détectés, nous évaluons les résultats en utilisant trois métriques à savoir le rappel, la précision et la f-mesure.

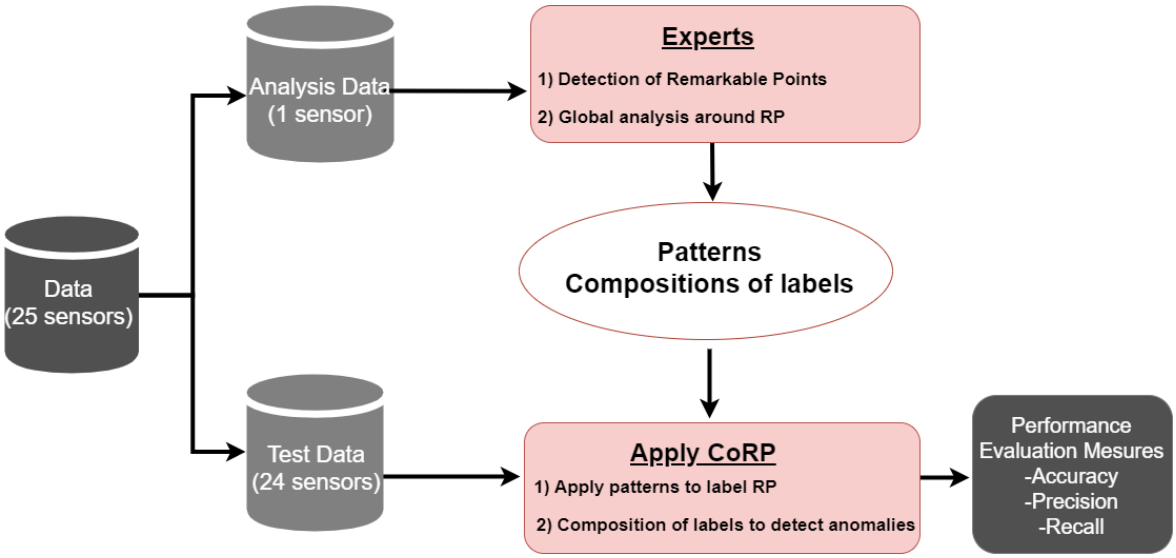


Figure III.2.1 – Processus d'évaluation de l'algorithme CoRP sur les données de SGE.

Les expériences ont été réalisées sur une machine Windows 10 Professional avec un processeur Intel (Core) i5 et 16 Go de RAM. Nous avons utilisé la distribution open source Python 3.7 Anaconda pour développer notre algorithme et R 3.5 pour explorer les algorithmes de la littérature.

2.3 Expérimentation sur des données réelles du SGE (séries croissantes et séries variables)

Dans cette partie, nous présentons une évaluation des méthodes suivantes : Règle Courte (noté SR), Règle Constante (noté CR), LOF, ARIMA, S-H-ESD et Change Point (noté LS). Nous avons appliqué ces méthodes par catégorie d'anomalies comme indiqué dans le tableau III.2.1. Afin d'évaluer leurs performances, nous utilisons, dans un premier temps, le nombre de vrais positifs (vraies anomalies détectées), le nombre de faux positifs (fausses anomalies détectées) et le nombre de faux négatifs (vraies anomalies non détectées) en tant que métriques d'évaluation. Par la suite, nous utilisons les mesures proposées dans la section I.2.7.2 (rappel, précision, F-mesure).

Comme les méthodes à évaluer ne sont pas entièrement automatisées, nous avons défini les valeurs de leurs paramètres tels que le seuil pour la Règle Courte, le nombre de voisins et le seuil pour évaluer le score du degré d'anomalie pour LOF ou le type

de modèle pour ARIMA, etc. Pour l'algorithme LOF, nous avons fait varier le choix du paramètre K, le nombre de voisins, dans une plage de 30 à 10 afin d'évaluer son influence sur le résultat de la détection (cf. haut de la figure III.2.2 B) et nous avons déterminé un seuil = 1,5 qui correspond à une distribution standard. Nous présentons les résultats de LOF dans un graphique séparé pour plus de lisibilité comme présenté dans la figure III.2.2 B.
Pour ARIMA, nous avons gardé les valeurs par défaut des paramètres (types d'anomalies, modèle ARIMA) défini dans le package.
Concernant la Règle Courte (SR), nous avons défini la valeur du seuil en utilisant la méthode basée sur l'histogramme décrite dans la section 2. Enfin, pour la règle constante (CR), nous avons fait varier le choix de la taille de la fenêtre coulissante dans une plage de 30 à 10 comme le montre la figure III.2.2 D (cf. haut de la figure III.2.2 D).

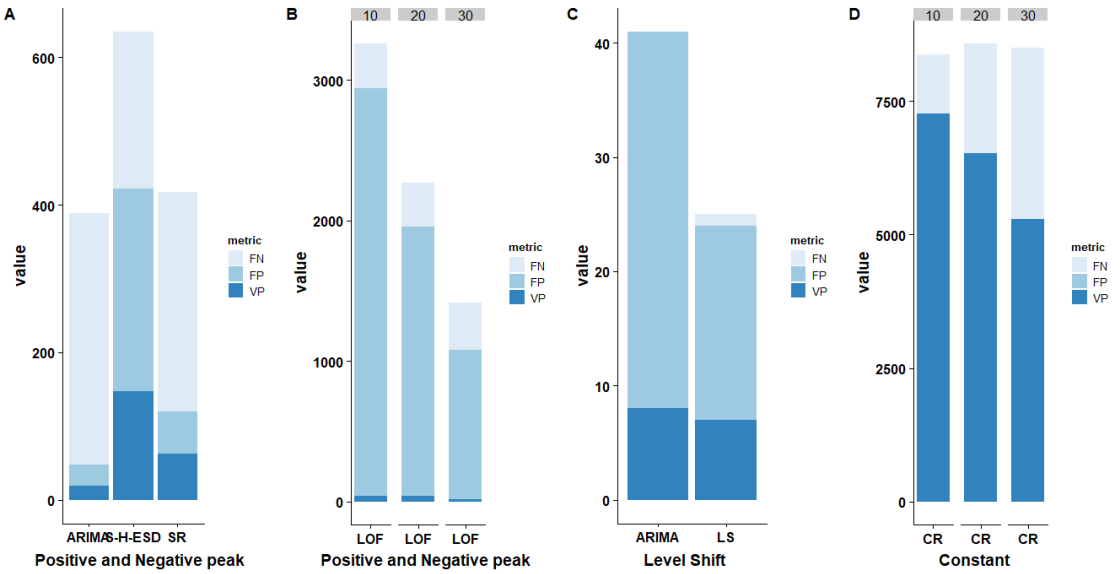


Figure III.2.2 – Évaluation des méthodes de détection d'anomalies sur les données d'index de calorie.

En se basant sur les résultats présentés dans les sous-figures III.2.2 A, III.2.2 B, III.2.2 C, et III.2.2 D, qui concernent les données d'index de calorie, nous pouvons faire les observations suivantes : LOF est la méthode qui génère le plus grand nombre de faux positifs tandis ARIMA génère le plus de faux négatifs. La méthode S-H-ESD est celle qui permet de détecter le plus de vrais positifs par rapport à LOF, ARIMA et SR, par contre elle engendre beaucoup de faux positifs et de faux négatifs. La Règle Courte (SR) détecte moins d'anomalies en comparaison avec S-H-ESD. Cependant, il en résulte moins de faux positifs que les autres méthodes. Également, nous pouvons dire que le nombre de voisins égal à 20 est le choix le plus approprié pour détecter le plus grand nombre d'anomalies avec l'algorithme LOF. Cependant, pour la Règle de

Constante (CR), il est important d'utiliser une taille de fenêtre suffisamment petite pour gérer des données contenant plusieurs anomalies constantes, à savoir 10 comme indiqué sur III.2.2 D.

Tableau III.2.2 – Comparaison des méthodes de détection d'anomalies sur les données de calorie d'index et de consommation.

Données	**Séries croissantes (Index)**			**Séries variables (Consommation)**		
	Precision	Recall	F-measure	Precision	Recall	F-measure
SR	0.52	0.17	0.32	0.66	0.63	0.64
CR	**1**	0.80	0.88	**1**	0.72	0.83
LOF	0.022	0.12	0.022	0.39	0.78	0.52
S-H-ESD	0.34	0.40	0.36	0.41	0.80	0.54
ARIMA	0.30	0.07	0.11	0.66	0.25	0.36
LS	0.29	0.87	0.43	-	-	-
CoRP	**1**	**1**	**1**	**1**	**0.98**	**0.98**

Le tableau III.2.2 présente les résultats de ces méthodes en fonction de la précision, du rappel et de la F-mesure sur les données d'index (série croissante) et sur les données de calorie de consommation (série variable). Nous donnons aussi les résultats de notre méthode CoRP. Nous avons mené les expérimentations par catégorie d'anomalies comme montré dans le tableau III.2.1.

Concernant les données de calorie d'index, en se basant sur ce tableau, nous déduisons que : (i) l'efficacité de la Règle Constante (CR) ou de la méthode LOF dépend fortement du choix de la fenêtre glissante ou du nombre de voisins ; (ii) la méthode Change Point fonctionne bien lorsqu'il y a réellement un changement de niveau dans la série temporelle, mais cependant, en cas d'absence d'anomalie, sa précision est faible ; et (iii) entre la Règle Courte (SR), ARIMA et S-H-ESD, la Règle Courte (SR) est la plus précise et ARIMA est la moins efficace pour détecter un changement anormal. En comparant avec ces méthodes, notre algorithme CoRP arrive à détecter de multiples types d'anomalies avec une meilleure précision et un meilleur rappel alors que les autres algorithmes ne peuvent pas détecter ces multiples anomalies.

Pour évaluer davantage notre algorithme et le confronter aux méthodes de détection d'anomalies, nous avons utilisé les données de consommation du SGE. Nous avons donc pris les mesures provenant des 24 capteurs (cf. la figure III.2.1). Les données de consommation sont des données saisonnières et leur évolution quotidienne, contrairement aux données d'index, est variable. Pour CoRP, nous avons inspecté manuellement un ensemble de données de même type, 1 série temporelle comme illustré dans la figure III.2.1 (la phase des experts), pour comprendre leurs variations. Ceci nous permet de

créer les motifs de détection des points remarquables qui peuvent exister et les compositions de labels pour détecter les anomalies. Les anomalies observées dans ces données sont les suivantes : pics positifs et négatifs, anomalies constantes, anomalies constantes commençant et se terminant par un décalage important. Ainsi, toujours avec l'aide des experts, nous avons créé 9 motifs afin de détecter les points remarquables et 5 compositions de labels pour détecter les anomalies.

Nous rapportons les résultats des algorithmes sur les données de consommation dans le tableau III.2.2 (séries variables). Comme les données ne sont pas stationnaires, nous n'avons pas appliqué l'algorithme Change Point parce qu'il n'existe pas de changement de niveau dans ces données à détecter. Ce tableau montre que les algorithmes de la littérature sont beaucoup plus efficaces sur les données de consommation en comparant avec les résultats sur les données d'index. Mais même sur ce type de données, notre approche a obtenu le meilleur résultat de F-mesure en comparant avec les autres algorithmes. En effet, CoRP a détecté le plus d'anomalies avec le moins d'erreurs possibles, avec une précision égale à 1 et un rappel égal à 0,98. Notons que, les résultats de la méthode à base de règles (SR, CR) et la méthode ARIMA ont une meilleure précision par rapport à LOF et SH-ESD. Enfin, SH-ESD est la méthode la plus proche du meilleur résultat en terme de rappel avec une valeur égale à 0,80. Toutefois, il faut noter que ces algorithmes n'arrivent pas détecter tous les types d'anomalies observées dans les déploiements réels, ce qui signifie que chaque algorithme est efficace dans un type spécifique. La particularité de notre méthode est que nous pouvons définir les motifs en fonction de nos besoins afin de détecter avec une grande précision et efficacité de multiples anomalies.

2.4 Expérimentation sur des données de la littérature (séries variables)

Afin d'évaluer l'algorithme dans un autre contexte, nous avons utilisé les ensembles de données de la méthode ARIMA (IPI et HIPC). Nous avons analysé manuellement un premier sous-ensemble de séries afin de spécifier les motifs en définissant un motif différent par type d'anomalies pour labelliser les points remarquables dans la série temporelle (3 motifs). Ensuite, nous avons procédé à une composition de ces labels pour détecter les anomalies (4 compostions de labels). Nous avons utilisé un deuxième sous-ensemble comportant deux séries temporelles de ces deux ensembles de données pour mener les expérimentations. Toutes les expérimentations sont faites par types d'anomalies.

Tableau III.2.3 – Comparaison de méthodes de détection d'anomalies sur des données de benchmark.

Datasets	**HIPC**		**IPI**	
Evaluation / Algorithm	Precision	Recall	Precision	Recall
ARIMA	**1**	**1**	**1**	**1**
LOF	0.11	0.20	0	0
S-H-ESD	0.20	0.20	0.33	0.25
SR	0	0	0	0
LS	0	0	0	0
CoRP	**1**	**0.80**	**0.75**	**0.75**

Le tableau III.2.3 est une comparaison entre les algorithmes de la littérature et notre algorithme sur les données proposées dans le package ARIMA. Nous n'avons pas testé la Règle Constante (RC) dans les ensembles de données HIPC et IPI car les anomalies observées dans ces données ne contiennent pas ce type d'anomalie. Nous avons par conséquent appliqué CoRP, ARIMA, LOF avec un nombre de voisins égal à 20, S-H-ESD, Change Point (LS) et la Règle Courte (SR) sur ces données. L'algorithme basé sur la Règle Courte (SR) et Change Point (LS) sont les moins précis parmi ces algorithmes, tandis que notre algorithme est le meilleur parmi eux et peut détecter la majorité des anomalies observées avec peu d'erreurs. Pour Change Point (LS) les valeurs sont mauvaises parce que l'algorithme n'identifie pas correctement les points d'arrêt ou les changements structurels. En effet, un point de changement est une instance dans le temps où les propriétés statistiques avant et après ce point temporel sont différentes. L'algorithme est efficace en cas de changements potentiels. Cependant, les changements dans les données IPI et HIPC ne sont pas si important pour que l'algorithme (LS) arrive à les détecter correctement. Pour la règle courte, le choix du seuil est défini en calculant le mode de l'histogramme des valeurs. Ensuite la différence entre chaque points successifs est comparée avec le seuil. Ce mode de définition de seuil est s'avère peu adapté aux données de ARIMA.

2.5 Conclusion

Ce chapitre présente l'approche CoRP basée sur des motifs appliqués aux séries temporelles uni-variées de données de capteurs. Notre méthode est composée de deux étapes : elle marque tous les points remarquables présents dans la série temporelle sur la base de motifs de détection, puis, elle identifie précisément les anomalies multiples présentes par compositions de labels. Cette approche nécessite l'expertise du domaine

d'application pour pouvoir définir efficacement les motifs et les compositions de labels. A l'inverse, les méthodes de la littérature, bien que moins efficaces, ne demandent pas autant d'expertise métier pour être appliquées.

Notre expérimentation est basée sur un contexte réel : les données de capteurs du SGE (service de gestion et d'exploitation du campus de Rangueil à Toulouse). L'évaluation de cette méthode est illustrée en utilisant tout d'abord les données d'index et de consommation des capteurs de calories exploités par le SGE et, en second lieu, en utilisant des jeux de données issus de l'état de l'art. Nous comparons notre algorithme à cinq méthodes appartenant à différentes techniques de détection d'anomalies. Sur la base des critères d'évaluation précision, rappel, f-mesure, nous montrons que notre algorithme est le plus efficace pour détecter différents types d'anomalies observées lors de déploiements réels en minimisant les fausses détections.

3 Expérimentation de la méthode CDT pour la génération des règles

"No one believes an hypothesis except its originator, but everyone believes an experiment except the experimenter."

William Ian Beardmore Beveridge (1908 — 2006)

Table des matières

3.1 Introduction

Notre deuxième contribution, présentée dans la section II.3 permet de créer un arbre de décision basé sur des compositions de motifs afin de générer automatiquement des règles de détection d'anomalies compréhensibles par les experts. L'intérêt de cette méthode est double. D'une part, elle permet de détecter des multiples anomalies efficacement. D'autre part, elle produit des règles intelligibles par les experts qui peuvent éventuellement les ajuster (fusion de règles, modification,...). Enfin, elle est automatique et ne requiert pas une configuration manuelle des paramètres.

Nous présentons les résultats des expérimentations réalisées afin de comparer nos résultats avec des méthodes de la littérature basées sur les motifs dans un premier temps, et sur des méthodes basées sur les règles dans un second temps. Ces expérimentations ont été menées sur les données réelles du SGE et les ensembles de données de référence de Yahoo Laptevand et Amizadeh (2015) présentés dans le chapitre III.1. Le code source de l'approche CDT et les ensembles de données SGE sont accessibles au public afin qu'ils soient facilement reproductibles. Pour cela, nous avons construit un notebook, qui contient le code utilisé dans ce travail [1]. Ce chapitre est organisé comme suit : nous exposons tout d'abord le protocole d'expérimentation que nous avons conçu avant de décrire les résultats obtenus. Puis nous présentons les règles produites par l'algorithme CDT. Enfin, nous conclurons le présent chapitre par une discussion et un bilan des expérimentations.

3.2 Protocole d'expérimentation

3.2.1 Processus d'évaluation

Comme nous l'avons décrit dans la section II.3.3, notre approche est basée sur cinq étapes. Nous illustrons le processus d'évaluation de ces étapes dans la figure III.3.1. Tout d'abord, les séries temporelles sont prétraitées en commençant par la normalisation, puis le sous-échantillonnage (downsampling) qui est une tâche optionnelle et qui dépend des données. Ensuite, nous appliquons le processus d'étiquetage automatique sur les séries avec des motifs. Enfin, nous appliquons le principe de fenêtres glissantes afin de préparer les observations pour CDT. En se basant sur les données d'apprentissage, nous créons notre arbre de décision afin d'avoir un modèle initial. Ce modèle sera évalué sur des données de validation en évaluant la fonction objectif de l'optimisa-

1. https ://github.com/IBK-TLS/CDT

tion bayésienne. Ceci permet de trouver les meilleurs hyper-paramètres du modèle qui maximise la fonction objectif. La dernière étape consiste à appliquer le modèle final sur les données de test, évaluer les performances de classification et générer les règles de détection d'anomalies.

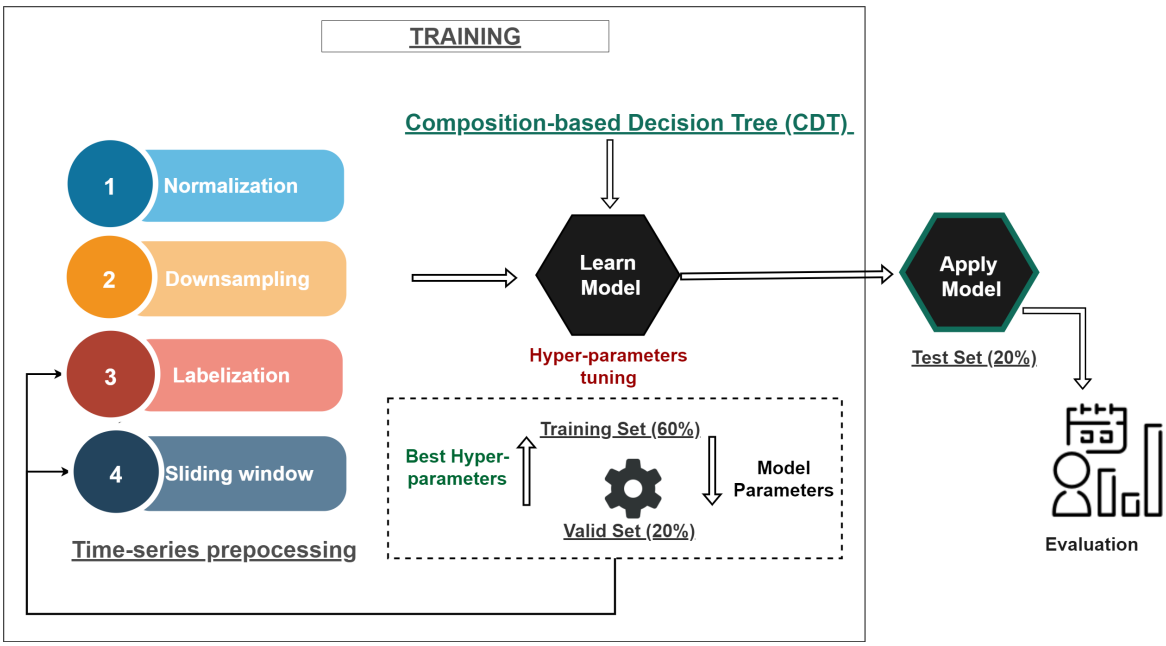


Figure III.3.1 – Processus d'évaluation de CDT

Nous évaluons CDT sur 6 ensembles de données à savoir les données de calorie et d'électricité de consommation du SGE et les 4 catégories des données de Yahoo (cf. le chapitre III.1). Le tableau III.3.1 donne la répartition de données en apprentissage (60%), validation (20%) et test (20%), et les anomalies présentes dans chaque dataset.

Tableau III.3.1 – Caractéristiques des ensembles de données utilisés dans les expérimentations.

Dataset	Train	Valid	Test	Total length	Anomalies	Δ_t
SGE_Electricity	52022	17819	26233	96074	10343 (10.77%)	1 h
SGE_Calorie	22520	5422	5594	33536	586 (1.75%)	1 day
Yahoo_A1	58977	15757	20108	94778	1669 (1.76%)	1 h
Yahoo_A2	85260	28420	28322	142002	466 (0.33%)	1 h
Yahoo_A3	100800	33600	33600	168000	943 (0.56%)	1 h
Yahoo_A4	100800	33600	33600	168000	837 (0.50%)	1 h

3.2.2 Mesure d'évaluation

Dans nos expérimentations, les performances de toutes les méthodes sont comparées en utilisant la F-mesure ou (F1 score), la métrique de qualité des règles $\mathcal{Q}$ présenté dans la section II.3.3.5 et de la fonction objectif $F(h)$ définie dans l'équation (3.10). Premièrement, nous utilisons $F1$ score pour comparer la précision de notre méthode par rapport aux méthodes basées sur les motifs. Deuxièmement, nous utilisons le score $F(h)$ pour évaluer à la fois la précision et l'interprétabilité des règles générées par notre algorithme CDT en comparant ce dernier avec les méthodes d'apprentissage de règles.

Pour l'évaluation, nous avons divisé chaque ensemble de données en trois sous-ensembles : ensemble d'apprentissage (60%), ensemble de validation (20%) et ensemble de test (20%), comme illustré dans la figure III.3.1. Nous utilisons les données d'apprentissage et de validation pour optimiser les valeurs des hyper-paramètres de notre modèle à l'aide de l'optimisation bayésienne. En effet, nous construisons l'arbre avec le jeu de données d'apprentissage, puis nous utilisons le jeu de données de validation pour évaluer la qualité prédictive de l'arbre avec les valeurs d'hyper-paramètre d'entrée. Ce processus est répété jusqu'à trouver un modèle stable avec les meilleurs hyper-paramètres. Finalement, nous évaluons le modèle final en utilisant le jeu de données de test.
Pour les méthodes basées sur les règles, supervisées, nous utilisons 80% des données pour l'apprentissage et 20% pour le test. Pour les méthodes basées sur les motifs, non supervisées, nous utilisons l'ensemble des données.

Optimisation des hyper-paramètres Afin de limiter l'espace de recherche de l'optimisation bayésienne, nous avons définit la plage de valeurs à chercher pour le paramètre ω entre [3,31]. Le minimum doit être 3 parce que une composition doit prendre en compte 3 points successifs (par définition de motif) et maximum 31 pour chercher des règles sur un mois au maximum. La plage de valeurs pour le paramètre δ est entre [1,21]. Ce choix est fait suite aux expérimentations. Nous supposons que un nombre de division δ au delà de cet intervalle va affecter la lisibilité des règles.

Le tableau III.3.2 montre les hyper-paramètres optimaux trouvés avec l'optimisation bayésienne. Comme nous pouvons le voir à travers le tableau III.3.2, l'optimisation sur $F(h)$ tend à favoriser un petit nombre de divisions (split) (δ) pour les motifs par rapport à l'optimisation du score $F1$. Cela est dû à la mesure de qualité des règles $\mathcal{Q}(\mathcal{R})$, qui favorise des règles courtes incluant un nombre minimum de motifs (δ). Cependant, la taille des observations (ω) nécessaires pour construire un arbre optimal reste comparable pour $F1$ et $F(h)$ pour les jeux de données SGE_Electricity

et Yahoo_A2, suggérant la nécessité de la présence d'un voisinage normal entourant une anomalie pour obtenir une bonne détection d'anomalies avec CDT.

Tableau III.3.2 – Hyper-paramètres de CDT pour l'expérimentation.

Evaluation	**F1-score**		**F(h)-score**	
Dataset	ω	δ	ω	δ
SGE_Electricity	27	2	27	2
SGE_Calorie	5	4	21	1
Yahoo_A1	27	16	25	1
Yahoo_A2	17	2	17	1
Yahoo_A3	29	12	17	1
Yahoo_A4	25	8	21	1

3.3 Expérimentation avec des algorithmes basés sur les motifs pour la détection d'anomalies

Nous utilisons les trois approches (discutées dans la section I.2) comme méthodes de base pour comparer avec notre méthode CDT : PBAD (Feremans *et al.*, 2019), MP (Yeh *et al.*, 2016) et PAV (Chen et Zhan, 2008).

— La détection des anomalies basée sur les motifs (PBAD) est une méthode de détection des anomalies basée sur des techniques d'exploration de modèles fréquents dans des séries chronologiques de type mixte.

— Matrix Profile (MP) est une mèthode qui détecte les motifs réguliers et les motifs anormaux dans une série temporelle. Pour cela, elle calcule les distances euclidiennes entre deux motifs standardisés. Les anomalies sont les discordes (sous-séquences inhabituelles) de séries chronologiques.

— Pattern Anomaly Value (PAV) est un algorithme de détection d'anomalies basé sur la valeur d'anomalie de motif. Les anomalies sont les motifs linéaires peu fréquents.

Pour évaluer ces méthodes, nous avons utilisé l'implantation disponible dans Feremans *et al.* (2019). Ces algorithmes sont des approches basées sur des fenêtres. Par conséquent, nous avons utilisé les paramètres recommandés pour chacun d'eux. Ainsi, nous avons divisé la série chronologique en fenêtres glissantes de longueur 12 avec un pas de 6 (Feremans *et al.*, 2019). Le résultat de ces algorithmes est un score d'anomalies pour chaque fenêtre. Comme ces algorithmes ne sont pas supervisés, nous construisons le modèle de détection d'anomalies sur l'ensemble complet des séries chronologiques et

Tableau III.3.3 – Évaluation de la détection d'anomalie en utilisant F1-score.

Dataset \ Algorithm	CDT	PBAD	PAV	MP
SGE_Electricity	**0.76**	0.70	0.74	0.70
SGE_Calorie	0.85	0.80	0.88	**0.91**
Yahoo_A1	**0.92**	0.72	0.75	0.76
Yahoo_A2	**0.99**	0.65	**0.99**	0.76
Yahoo_A3	**1.0**	0.73	0.99	0.70
Yahoo_A4	**0.98**	0.75	0.93	0.96
Average	**0.92**	0.72	0.88	0.80
Min	**0.76**	0.65	0.74	0.70
Max	**1.0**	0.80	0.99	0.96

nous l'évaluons à l'aide du score $F1$. Concernant CDT, nous avons utilisé les valeurs appropriées des hyper-paramètres calculés à l'aide du score F1 comme indiqué dans le tableau III.3.2.

Tous les jeux de données sont normalisés entre 0 et 1 lors de la phase de pré-traitement. Nous appliquons la normalisation par série/fichier.
Pour les jeux de données Yahoo et SGE-Electricity, une anomalie collective peut durer près de deux jours. Pour garantir un voisinage normal autour d'un motif, nous avons sous-échantillonné ces ensembles de données d'heures en jours. Ce choix a été appliqué pour CDT et pour les compétiteurs.

Analyse des résultats Le tableau III.3.3 montre le score $F1$ obtenu par chaque algorithme sur chacun des six ensembles de données de séries chronologiques uni-variées. Les meilleures valeurs du $F1$ score pour chaque ensemble de données sont indiquées en gras. Nous calculons également le rang moyen de chaque méthode, le minimum ainsi que le maximum. Comme illustré dans le tableau, CDT surpasse les compétiteurs sur cinq parmi six ensembles de données.
On peut observer dans le tableau III.3.3 que notre méthode est plus stable et cohérente pour différents ensembles de données par rapport aux algorithmes de la littérature. Nous remarquons également que la précision de CDT dépasse les autres avec +0.16 sur le dataset Yahoo_A1. En moyenne, sur tous les ensembles, CDT dépasse PAV de 0.04, MP de 0.12 et PBAD de 0.20. Le F1-score minimum de CDT, 0.74, est le meilleur par rapport aux autres algorithmes ainsi que le maximum avec une valeur de 1.0. Donc dans l'ensemble, CDT n'est jamais le moins bon en comparant avec les compétiteurs.
La précision de MP est très variable d'un ensemble de données à l'autre étant donnée

qu'elle dépend des types d'anomalies similaires dans les ensembles de données. En effet, elle détecte les fenêtres anormales qui ont la plus grande distance par rapport à la fenêtre voisine la plus proche.

Pour les algorithmes concurrents, les données doivent être équilibrées sinon les résultats de détection sont médiocres. Nous avons donc testé PBAD, PAV et MP sur nos jeux de données initiaux et sur une version équilibrée et nous avons constaté que, leurs performances se dégradent fortement dans le premier cas. Ceci est lié au fait qu'ils ont tendance à se concentrer sur l'exactitude des prédictions de la classe majoritaire (classe normale) qui génère une faible précision pour la classe minoritaire (classe d'anomalies). Nous avons appliqué notre modèle à la fois dans des ensembles de données équilibrés et non équilibrés et nous avons constaté qu'il atteint la plus grande précision de détection sur des ensembles de données sans prétraitement (non équilibrés). Les résultats de CDT présentés dans le tableau III.3.3 correspondent à l'évaluation sur les données sans prétraitement.

3.4 Expérimentations avec des algorithmes d'apprentissage de règles

Les algorithmes d'apprentissage de règles génèrent un modèle sous la forme d'un ensemble de règles. Ce modèle prédictif est facilement interprétable, et ne nécessite pas de connaissances statistiques préalables. Les règles sont sous la forme standard de règles de type « IF-THEN » ou « Si condition Alors Conclusion ».

Nous comparons notre méthode CDT avec les algorithmes d'apprentissage de règles suivants implantés dans WEKA (discutées dans la section I.2).

— JRip, qui correspond à RIPPER de Cohen (1995), implémente un apprentissage de règles et un élagage incrémental pour produire une réduction des erreurs (RIPPER). Elle est disponible sous l'appellation JRIP dans le logiciel Weka. L'algorithme est basé sur l'extraction des règles d'associations séquentielles fréquentes dont la partie droite est une anomalie. JRIP produit des règles indépendantes. La méthode intègre une première procédure de post-élagage basée sur la description minimale des messages pour raccourcir les règles en retirant les propositions inutiles, et une seconde procédure pour réduire le nombre de règles dans la base.

— PART (Frank et Witten, 1998) est une combinaison des algorithmes d'apprentissage de règles C4.5 et de RIPPER pour produire des règles à partir d'arbres de décision partiels en utilisant l'algorithme C4.5. L'algorithme génère des listes de

Tableau III.3.4 – Évaluation de la détection d'anomalies en utilisant $F1$ score, la mesure de qualité $\mathcal{Q}(\mathcal{R})$ et la fonction objectif $F(h)$.

Evaluation	**F1-score**			**Q(R)**			**F(h)-score**		
Algorithm / Dataset	CDT	PART	JRip	CDT	PART	JRip	CDT	PART	JRip
SGE_Electricity	**0.76**	0.71	0.72	0.67	0.67	**0.70**	**0.51**	0.48	0.50
SGE_Calorie	**0.99**	0.80	0.79	0.61	0.65	**0.69**	**0.60**	0.52	0.54
Yahoo_A1	**0.91**	0.70	0.69	0.48	0.50	**0.56**	**0.43**	0.35	0.39
Yahoo_A2	**0.99**	0.80	0.77	**0.69**	0.68	0.65	**0.68**	0.54	0.50
Yahoo_A3	**0.98**	0.78	0.71	**0.77**	0.69	0.70	**0.75**	0.54	0.50
Yahoo_A4	**0.97**	0.73	0.75	**0.70**	**0.70**	0.68	**0.68**	0.51	0.51
Average	**0.93**	0.75	0.74	**0.65**	0.64	0.64	**0.61**	0.49	0.49
Min	**0.76**	0.70	0.69	0.48	0.50	0.56	**0.43**	0.35	0.39
Max	**0.99**	0.80	0.79	**0.77**	0.70	0.70	**0.75**	0.54	0.54

décision. La méthode consiste à créer un arbre de décision à chaque étape, sélectionner la branche la plus intéressante, retirer les observations associées, et réitérer le processus jusqu'à épuisement de la base.

Nous mettons en oeuvre ces méthodes avec le logiciel WEKA (Waikato Environment for Knowledge Acquisition) dans sa version 3.8 (Witten et Frank, 2005). WEKA est logiciel open-source qui intègre toute une panoplie de méthodes d'apprentissage automatique telles que la classification, le regroupement et les règles d'association et beaucoup d'autres.

Pour la construction et l'évaluation des algorithme PART et JRIP, nous avons gardé la configuration par défaut des paramètres de WEKA et nous avons utilisé la validation croisée des plis en K. La validation croisée va nous permettre d'utiliser l'intégralité de notre jeu de données pour l'entraînement et pour la validation. Le principe est de découper aléatoirement le jeu de données en k parties (folds en anglais) à peu près égales. Tour à tour, chacune des k parties est utilisée comme jeu de test. Le reste (autrement dit, l'union des k-1 autres parties) est utilisé pour l'entraînement. Nous rapportons finalement la performance du modèle en moyennant les performances obtenues sur les k folds. Dans nos tests, sur WEKA, le k est égale à 10 pour JRip et PART. En ce qui concerne CDT, nous avons appliqué le processus d'évaluation présenté dans la section III.3.2.1.

Pour CDT et chacun des compétiteurs, nous avons utilisé les valeurs des hyper-paramètres obtenues par l'optimisation bayésienne dans CDT pour maximiser $F(h) -$

score comme montré dans le tableau III.3.2.

Analyse des résultats Tout d'abord, nous comparons CDT avec PART et JRip en utilisant $F1$ score, $\mathcal{Q}(\mathcal{R})$ et $F(h)$ score comme le montre le tableau III.3.4). Ensuite nous les comparons en fonction du nombre de règles générées par chacun d'entre eux comme illustré dans la figure III.3.2.
Le tableau III.3.4 présente une comparaison des résultats de chaque algorithme sur les six ensembles de données en utilisant $F1$, $\mathcal{Q}(\mathcal{R})$ et $F(h)$ score. Dans l'ensemble, les scores moyens montrent que notre approche a obtenu la première position du classement suivie de PART et JRip.

CDT surpasse PART et JRip dans cinq des six ensembles de données avec la mesure $F1$ score, dans trois des six ensembles de données ave la mesure $\mathcal{Q}(\mathcal{R})$ et sur tous les ensembles de données avec la mesure $F(h)$ score.

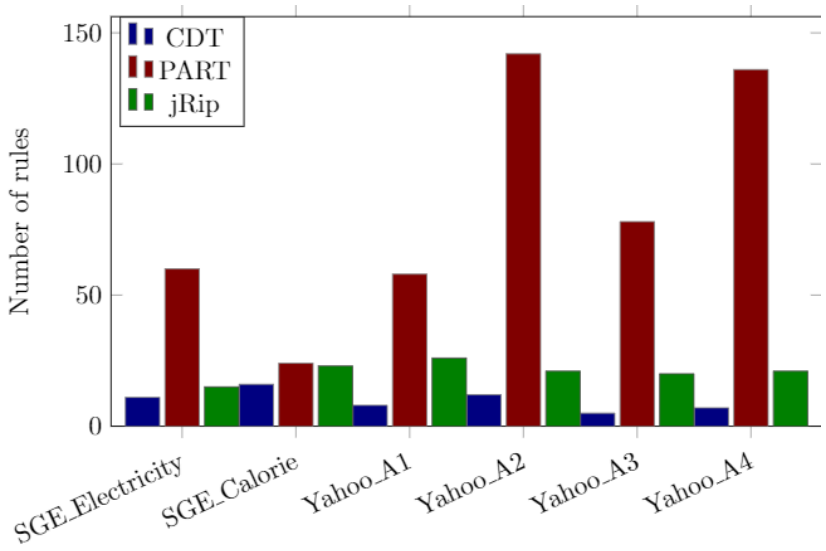


Figure III.3.2 – Le nombre de règles générées pour pour les algorithmes CDT, PART et JRip pour la détection d'anomalies.

Nous pouvons constater à partir du tableau III.3.4 que les performances des algorithmes de règles varient en fonction du nombre d'attributs. Typiquement, avec $\omega = 31$ pour l'ensemble de données Yahoo_A1, PART et JRip obtiennent un score $F1$ inférieur à celui du reste des ensembles de données. Nous pouvons observer à partir du tableau III.3.4 que JRip a une haute qualité de règles $\mathcal{Q}(\mathcal{R})$ dans trois ensembles de données ainsi que CDT. Cela est dû à la taille de ses règles générées qui sont assez courtes. Cependant, il est moins précis que CDT et PART dans presque tous les ensembles de données. Nous pouvons également voir qu'aucun des algorithmes compétiteurs obtient un bon $F(h)$ score sur tout les ensembles de données. Alors que CDT a le meilleur

compromis entre le score $F1$ et le score $\mathcal{Q}(\mathcal{R})$.

La figure III.3.2 montre un résumé du nombre de règles produites par chaque méthode. CDT produit peu de règles entre 5 et 16 règles. Ce nombre réduit de règles est obtenu grâce au principe de la recherche des compositions de motifs. CDT est suivi par JRip qui a produit raisonnablement peu de règles entre 15 et 30 règles. Ceci est lié au processus d'élagage appliqué dans l'algorithme. En fin, PART produit beaucoup de règles, entre 24 et 142. Cela est dû à la spécificité des règles générées qui ont un faible support.

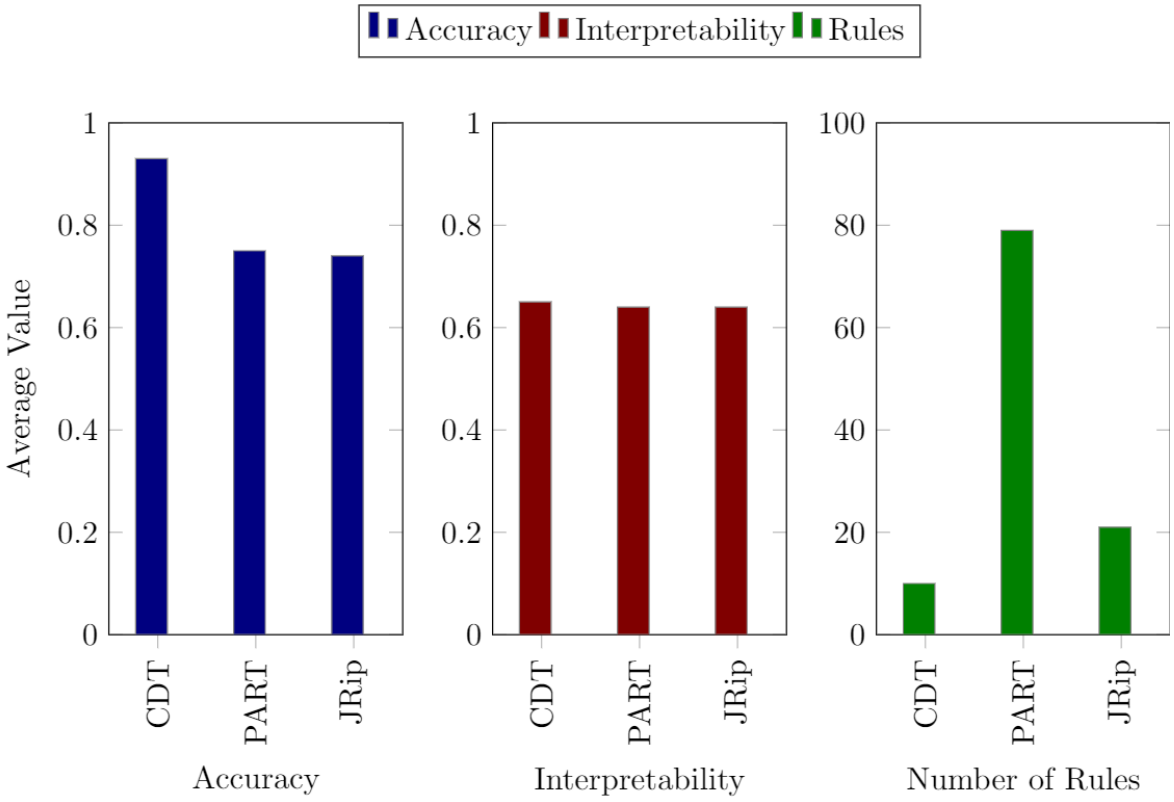


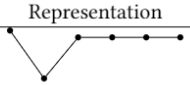
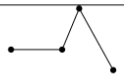
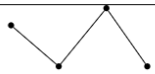
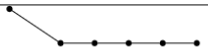
Figure III.3.3 – Le compromis entre la précision, l'interprétabilité et le nombre de règles pour les algorithmes CDT, PART et JRip.

Nous résumons les résultats moyens de tous les algorithmes dans la Figure III.3.3, montrant le compromis entre la précision, la qualité et le nombre de règles générées. De la figure III.3.3, nous pouvons conclure que CDT est le classificateur le plus précis qui produit des règles interprétables moins nombreuses, tandis que PART a une bonne précision par rapport à JRip mais produit un nombre important de règles.

Exemple de règles produites Nous présentons quelques exemples de règles générées par notre algorithme CDT à partir d'ensembles de données du SGE de calorie pour détecter de multiples anomalies dans le tableau III.3.5. Les règles avec des motifs

visualisés sont faciles à comprendre et peuvent être intuitivement interprétées par les utilisateurs. Elles sont décrites comment suit : le pic négatif (le point à la position 2 dans la représentation du tableau III.3.5) est considéré comme une anomalie car la consommation d'énergie dans un bâtiment ne peut pas être négative. Le pic positif (le point à la position 3 dans la représentation du tableau III.3.5) est survenu suite à une surconsommation dans le bâtiment. Les anomalies de bruit présentent des variations anormales en points successifs (les points à la position 2 et 3 dans la représentation du tableau III.3.5). Cela est dû à un défaut de lecture des compteurs. Enfin, l'anomalie constante illustre un arrêt du compteur (tous les points à partir de la position 2 dans la représentation du tableau III.3.5).

Tableau III.3.5 – Exemple de règles générées par CDT pour la détection d'anomalies sur les données de calorie du SGE.

Rule Predicate	Support	Representation	Type of anomaly	Interpretation
$[PN_{-H,-H}, SCP_{-H,0}, CST, CST, CST]$ and $\neg\,[ECN_{0,-H}, SCN_{-H,0}]$	344		negative peak	error of measure of the meter
$[ECP_{0,-L}, PP_{L,H}]$	55		positive peak	overconsumption of energy
$[PN_{-H,-H}, PP_{H,H}]$	8		noise anomaly	reading error of the meters
$[SCN_{H,0}, CST, CST, CST, CST, CST, CST, CST]$ and $\neg\,[PN_{-H,-H}, SCP_{H,0}, CST]$	4		constant anomaly	stop of the meter

3.5 Conclusion

Notre méthode CDT est une méthode d'apprentissage automatique qui peut générer des règles intelligibles par les experts du domaine pour la détection d'anomalies multiples dans les séries temporelles. L'approche est basée sur la labélisation automatique des séries temporelles en utilisant des motifs. Compte tenu de cet étiquetage, un arbre de décision modifié est ensuite construit, en considérant les nœuds comme des compositions de motifs. En utilisant l'optimisation bayésienne, nous avons optimisé les hyper-paramètres de manière à maximiser à la fois la qualité des règles et les performances de classification. Les performances de la méthode présentée ont été testées à l'aide de l'ensemble de données Yahoo et du SGE et la faisabilité pratique a été évaluée avec des ensembles de données du monde réel (SGE).

Conclusion générale

« Une méthode fixe n'est pas une méthode. »

Proverbe chinois

Ces travaux de thèse ont été financés par le Service de Gestion et d'Exploitation (SGE) du campus de Rangueil rattaché au Rectorat de Toulouse et la recherche est menée dans le cadre du projet neOCampus (Université Paul Sabatier, Toulouse).

Synthèse des propositions

Dans le contexte de la supervision des réseaux de capteurs, nous avons exposé l'enjeu de la détection d'anomalies dans les séries temporelles. La détection d'anomalies dans des applications de distribution de fluides réelles est une tâche difficile, en particulier, lorsque nous cherchons à détecter avec précision différents types d'anomalies et d'éventuelles défaillances de capteurs. Le deuxième enjeu consiste à proposer des règles, compréhensibles par un expert, permettant de détecter finement une anomalie. La résolution de ce problème est de plus en plus importante dans les applications de gestion et de supervision des bâtiments pour l'analyse et la supervision. Notre étude de cas s'appuie sur un contexte réel : les données des capteurs du SGE (Service de gestion et d'exploitation du campus de Rangueil à Toulouse).

Afin de proposer une solution fiable à ces problématiques, nous avons exposé dans ce mémoire deux propositions :

— notre première contribution est une approche configurable basée sur la modélisation de motifs de détection d'anomalies multiples. Notre algorithme intitulé

CoRP (Composition of Remarkable Points), applique un ensemble de motifs, définis par l'expert, afin d'annoter les points remarquables dans une série temporelle uni-variée, puis détecte les anomalies par composition de labels. Concernant la validation de cette contribution, nous avons appliqué notre algorithme sur les bases de données du SGE afin de démontrer l'efficience de cette solution. Par ailleurs, nous avons établi un protocole d'expérimentations afin de comparer CoRP avec des algorithmes de la littérature. Ces expérimentations reposent sur des données du monde réel et des données de benchmarks issus de la littérature. Les résultats obtenus montrent que notre approche est plus robuste et précise pour détecter tous les types d'anomalies ;

— notre deuxième contribution est une modélisation automatisée des motifs de détection des points remarquables et une version adaptée des arbres de décision, appelée CDT (Composition-based Decision Tree), pour produire les règles. Notre approche permet de générer automatiquement des règles de décision pour la détection d'anomalies. Ces règles sont intelligibles et compréhensibles par les experts et les analystes qui peuvent les ajuster et les modifier. Cette approche n'impose pas de configuration manuelle des paramètres. En effet, à l'aide d'une optimisation bayesienne, nous cherchons automatiquement les meilleures valeurs des hyperparamètres afin de construire un modèle d'arbre performant. Cette contribution a fait l'objet également de validations expérimentales qui ont été menées sur des données réelles et synthétiques. Nous montrons que notre méthode est précise pour classifier les anomalies par rapport aux autres méthodes. Également, elle permet de générer des règles interprétables. Le prototype CDT peut être téléchargé et installé à partir du site Web dédié « `https://github.com/IBK-TLS/CDT` ».

Champs d'application de notre approche

Nous avons focalisé dans notre approche sur les séries temporelles uni-variées, notamment dans le contexte des réseaux de capteurs. Dans ce domaine, il est primordial de détecter les anomalies afin de remonter les alarmes aux experts et leurs faciliter la supervision et l'analyse des données. Ils peuvent ainsi, améliorer la qualité des données par suppression ou remplacement des données aberrantes ou encore avoir de nouvelles connaissances utiles au travers des anomalies détectées.

Même si elles ont été proposées dans le cadre de réseaux de capteurs, les contributions présentées dans ce mémoire peuvent se décliner dans divers autres contextes d'application où les séries temporelles sont présentes. Parmi les champs d'applica-

tion possibles, l'industrie (détection de défaillance ou défaut de fabrication), la finance (évolution des indices boursiers des données économique), la santé (analyse d'électroencéphalogrammes et suivi des patients), la science de la Terre (indices de marées), l'assurance (analyse des sinistres et détection de fraude), le trafic réseaux (analyses des attaques et des comportement malicieux), et bien d'autres domaines.

Perspectives de recherche

Certaines interrogations soulevées au long de la présentation des approches développées durant cette thèse ouvrent des pistes intéressantes pour de futurs travaux de recherche. Les travaux futurs comprennent, sans s'y limiter, les points suivants :

- Perspectives à court terme
  - Approfondir nos expérimentations en intégrant les paramètres que nous avons fixés par défaut comme le pas de la fenêtre glissante ou la valeur de sous-échantillonage dans l'optimisation bayésienne. De tels évaluations pourront être approfondies en évaluant les résultats et les gains qui en découlent sur la qualité des règles produites et sur la performance de classification.
  - Simplifier les règles produites à travers des heuristiques pour minimiser le nombre de compositions. Par exemple, supprimer les compositions par inclusion. Ceci permet d'éliminer les compositions redondantes dans une règle. Ou encore, vérifier le chevauchement entre des compositions positives et compositions négatives en fonction de la taille de la fenêtre.
- Perspectives à moyen terme
  - Rendre CDT multi classes afin de catégoriser les règles générées en fonction de la classe d'anomalies et avoir un aperçu clair sur les types d'anomalies trouvées. La version actuelle de CDT permet de générer un arbre binaire qui indique si une observation est normale ou anomalie. Une modification dans le paramétrage de CDT peut être réalisée pour supporter des classes multiples. Ceci demandent bien évidement des données avec des labels multi classes.
  - Chercher l'emplacement d'anomalies dans les règles générées par CDT. Ceci a pour but de simplifier davantage les règles et les rendre plus compréhensibles. Nous pourrons envisager d'appliquer CDT sur les valeurs numériques des séries temporelles et créer ainsi un arbre de décision hybride basé sur les motifs et sur les valeurs. De cette manière nous pouvons conserver une trace de la localisation d'anomalies dans les observations.

- Perspectives à long terme

  — Appliquer nos approches sur les séries temporelles multi-variées. Dans un premier temps, nous envisageons d'adapter notre arbre CDT afin de détecter les anomalies uni et multi-variées.

  Les anomalies uni-variées dans les séries temporelles multi-variées se produisent indépendamment d'un signal à l'autre. Par conséquent, un traitement séparé des différents signaux composant la série temporelle est approprié par CDT.

  Les anomalies multivariées se produisent sur plusieurs variables de manière plus au moins corrélées, par exemple parce que les capteurs mesurent des grandeurs de natures différentes (vitesse, pression, puissance électrique, etc.). Dans ce cas de figure, nous envisageons de pré-traiter les séries temporelles multivariées en utilisant la réduction des variables par exemple ou modifier le noyau de CDT pour supporter ce type de données.

Bibliographie

« Le but d'une lecture intelligente est votre instruction. Cela fera mieux que de vous aider à passer le temps ; la lecture changera la nature de vos relations avec autrui ; elle déterminera en vous des perceptions plus rapides, de nouveaux concepts et de nouvelles formes de pensée, car sa fonction principale est de vous éveiller. Et grâce à la lecture vous découvrirez en vous-même et dans le monde des possibilités nouvelles. »

Howard Phillips Lovecraft (1890 — 1937)

Xu, S. et Balazinska, M. (2011). Sensor Data Stream Exploration for Monitoring Applications. DMSN. Cité 2 fois, p. 1 et 3.

Cateni, S., Colla, V., Vannucci, M., Aramburo, J. et Trevino, A.R. (2008). Outlier detection methods for industrial applications. Advances in Robotics. *In Advances in Robotics, Automation and Control*, pages 265-282. IN-TECH. Cité 2 fois, p. 1 et 2.

Chakrabarti, A., Marwah, M. et Arlitt, M. (2016). Robust anomaly detection for large-scale sensor data. *In BuildSys'16: Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, pages 31-40, Palo Alto, CA, USA. ACM Press. Cité 1 fois, p. 22.

Chandola, V., Banerjee, A. et Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3), pages 1-58. Cité 13 fois, p. 2, 4, 19, 20, 21, 22, 24, 26, 27, 28, 50, 65 et 129.

Gupta, M., Gao, J., Aggarwal, C.C. et Han, J. (2013). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and data Engineering*, 26(9), 2250-2267. IEEE. Cité 3 fois, p. 2, 4 et 19.

Aggarwal, C.C. (2015). Outlier analysis. *In Data mining*, page 237-263. Springer. Cité 4 fois, p. 2, 4, 20 et 65.

Sharma, A.B., Golubchik, L. et Govindan, R. (2010). Sensor faults: Detection methods and prevalence in real-world datasets. *In TOSN'10: ACM Transactions on Sensor Networks (TOSN)*, 6(3), 1-39, New York, NY, USA. ACM Press. Cité 7 fois, p. 2, 25, 27, 29, 38, 50 et 94.

Kiani, R., Keshavarzi, A. et Bohlouli, M. (2020). Detection of Thin Boundaries between Different Types of Anomalies in Outlier Detection using Enhanced Neural Networks. *Applied Artificial Intelligence*, 34(5), 345-377. Cité 2 fois, p. 2 et 50.

Brockwell, P.J., Davis, R.A. et Calder, M.V. (2002). Introduction to time series and forecasting. Vol. 2, pp. 3118-3121, New York, NY, USA. springer. Cité 1 fois, p. 12.

Keogh, S., Chu, J., Hart, D. et Pazzani, M. (2004). Segmenting time series: A survey and novel approach. *In Data mining in time series databases*, PAGES 1-21. World Scientific. Cité 1 fois, p. 14.

Horst, B. et Abraham, K. (2004). Data mining in time series databases. Vol. 57, World scientific. Cité 1 fois, p. 14.

Archana, N. et Pawar, S.S. (2014). Robust Pointing by XPath Language: Authoring Support and Empirical Evaluation. *In IJSR'14: International Journal of Science and Research (IJSR)*, vol. 1, 1852-1856. Cité 1 fois, p. 12.

Zhou, C., Cule, B. et Goethals, B. (2015). Pattern based sequence classification. *IEEE Transactions on knowledge and Data Engineering*, vol. 28(5), pages 1285-1298, IEEE. Cité 1 fois, p. 14.

Cheng, H., Tan, P.N., Potter, C. et Klooster, S. (2009). Detection and characterization of anomalies in multivariate time series. *In Proceedings of the 2009 SIAM international conference on data mining*, pages 413-424. Society for Industrial and Applied Mathematics. Cité 1 fois, p. 13.

Witten, I.H., Frank, E, Hall, M.A. et Pal, C.J. (2016). Data Mining: Practical machine learning tools and techniques. Cité 1 fois, p. 15.

Hodge , V. et Austin, J. (2004). A survey of outlier detection methodologies. *In Artificial intelligence review*, vol. 22(2), pages 85-126, Springer. Cité 2 fois, p. 20 et 27.

Keogh, E., Lin, J., Lee, S.H. et Van Herle, H. (2007). Finding the most unusual time series subsequence: algorithms and applications. Knowledge and Information Systems, vol. 11(1), pages 1-27. Springer. Cité 1 fois, p. 20.

Carreno, A., Inza, I. et Lozano, J.A. (2019). Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. Artificial Intelligence Review, pages 1–20. Springer. Cité 1 fois, p. 19.

Hawkins, D.M. (1980). Identification of outliers. London: Chapman and Hall., Vol. 11. Springer. Cité 1 fois, p. 20.

Barai, A. et Dey, L. (2017). Outlier Detection and Removal Algorithm in KMeans and Hierarchical Clustering. World Journal of Computer Application and Technology, VOL. 5(2), pages 24–29. Cité 1 fois, p. 20.

Yao, Y., Sharma, A., Golubchik, L. et Govindan, R. (2010). Online anomaly detection for sensor systems: A simple and efficient approach. Performance Evaluation, vol. 67(11), pages 1059–1075. Elsevier. Cité 2 fois, p. 25 et 50.

Balke, N.S. (1993). Detecting level shifts in time series. Journal of Business & Economic Statistics, vol. 11(1), pages 81-92. Cité 1 fois, p. 25.

Zhang, Y., Meratnia, N. et Havinga, P. (2007). A Taxonomy Framework for Unsupervised Outlier Detection Techniques for Multi-Type Data Sets. Rap. tech., Centre for Telematics and Information Technology University of Twente. Cité 1 fois, p. 27.

Hori, S.S. (2014). A Survey on Outlier Detection Methods. International Journal of Computer Science and Information Technologies, vol.5(6), pages 8153-8156. Cité 2 fois, p. 29 et 38.

Breunig, M.M., Kriegel, H.P., NG, R.T. et Sander, J. (2000). Lof: identifying density-based local outliers. *In SIGMOD'00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, vol. 29, pages 93–104. , New York, NY, USA. ACM Press. Cité 3 fois, p. 34, 38 et 94.

Rosner, B. (1983). Percentage points for a generalized ESD many-outlier procedure. Technometrics, vol. 25(2), pages 165–172. Cité 4 fois, p. 25, 29, 30 et 95.

Basseville, D., Nikiforov, I.V. (1993). Detection of abrupt changes: theory and application. vol. 104. Englewood Cliffs: prentice Hall. Cité 3 fois, p. 25, 29 et 38.

Aminikhanghahi, S. et Cook, D.J (2017). A survey of methods for time series change point detection. Knowledge and information systems, vol. 51(2), pages 339–367. Springer. Cité 2 fois, p. 29 et 95.

Upadhyaya, S. et Singh, K. (2012). Nearest neighbour based outlier detection techniques. International Journal of Computer Trends and Technology, vol. 3(2), pages 299–303. Cité 3 fois, p. 25, 34 et 38.

Ozyildirim, B.M. et Avci, M. (2013). Generalized classifier neural network. Neural Networks, vol. 39, pages 18-26. Elsevier. Cité 1 fois, p. 30.

Rashidi, L., Hashemi, S. et Hamzeh, A. (2011). Anomaly Detection in Categorical Datasets Using Bayesian Networks. *In AICI'11: International Conference on Artificial Intelligence and Computational Intelligence*, pages 610–619. Springer, Berlin, Heidelberg. Cité 2 fois, p. 30 et 31.

Hejazi, M. et Singh, Y.P. (2013). One-class support vector machines approach to anomaly detection. Applied Artificial Intelligence, vol. 27(5), pages 351-366. Cité 1 fois, p. 30.

Duffield, N., Haffner, P., Krishnamurthy, B. et Ringberg, H. (2009). Rule-based anomaly detection on IP flows. *In IEEE INFOCOM 2009*, pages 424-432. IEEE. Cité 1 fois, p. 31.

Moayedi, M. et Masnadi-Shirazi, M. (2008). Arima model for network traffic prediction and anomaly detection. *In SAINT'03: 2008 International Symposium on Information Technology*, vol. 4, pages 1–6. IEEE. Cité 1 fois, p. 30.

Pena, E.H., de Assis, M.V. et Proença, M.L. (2013). Anomaly detection using forecasting methods arima and hwds. *In SCCC'13: 2013 32nd International Conference of the Chilean Computer Science Society (SCCC)*, pages 63-66. IEEE. Cité 1 fois, p. 30.

Zhu, B. et Sastry, S. (2011). Revisit dynamic arima based anomaly detection. *In PASSAT/SocialCom'11: 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 1263-1268. IEEE. Cité 1 fois, p. 30.

Chen, C. et Liu, L.M. (1993). Joint estimation of model parameters and outlier effects in timeseries. Journal of the American Statistical Association, VOL. 88(421), pages 284–297. Cité 4 fois, p. 25, 30, 38 et 95.

Gaddam, S.R., Phoha, V.V. et Balagani, K.S. (2007). K-means+id3: A novel method for supervised anomaly detection by cascading k-means clustering and ID3 decision tree learning methods. IEEE transactions on knowledge and data engineering, vol. 19(3), pages 345-354. IEEE. Cité 2 fois, p. 32 et 34.

Muniyandi, A.P., Rajeswari, R. et Rajaram, R. (2012). Network Anomaly Detection by Cascading k-Means Clustering and c4.5 Decision Tree algorithm. Procedia Engineering, vol. 30, pages 174-182. Elsevier. Cité 2 fois, p. 32 et 34.

Gupta, B., Rawat, A., Jain, A., Arora, A. et Dhami, N. (2017). Analysis of various decision tree algorithms for classification in data mining. International Journal of Computer Applications, vol 163(8), pages 15-19. Foundation of Computer Science. Cité 1 fois, p. 32.

Sinwar, D. et Kumar, M. (2016). Anomaly Detection using Decision Tree based Classifiers. *In IJMTER'16: International Journal of Modern Trends in Engineering and Research(IJMTER).* Cité 2 fois, p. 32 et 38.

Chen, D., Liaw, R. et Breiman, M. (2004). Using random forest to learn imbalanceddata. University of California, Berkeley, vol. 110, pages 1-12. Cité 1 fois, p. 33.

Hori, M. (2001). Random forests. *In* Machine learning, 45(1), pages 5-32. Springer. Cité 2 fois, p. 32 et 38.

Ma, J. et Perkins, S. (2003). Time-series novelty detection using one-class support vector machines. *In Proceedings of the International Joint Conference on Neural Networks*, Vol. 3, pages 1741-1745. IEEE. Cité 1 fois, p. 31.

Mukkamala, S., Janoski, G. et Sung, A. (2002). Intrusion detection using neural networks and support vector machines. *In IJCNN'02: Proceedings of the 2002 International Joint Conference on Neural Networks*, vol.2, pages 1702-1707. IEEE. Cité 1 fois, p. 31.

Javaid, A., Niyaz, Q., Sun, W. et Alam, M. (2016). A deep learning approach for network intrusion detection system. *In BIONETICS'16: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies*, pages 21-26. Cité 2 fois, p. 21 et 31.

Jadidi, Z., Muthukkumarasamy, V., Sithirasenan, E. et Sheikhan, M. (2013). A deep learning approach for network intrusion detection system. *In ICDCS'13: 2013 IEEE 33rd international conference on distributed computing systems workshops*, pages 76-81. IEEE. Cité 2 fois, p. 21 et 31.

Münz, G., Li, S. et Carle, G. (2007). Traffic Anomaly Detection Using KMeans Clustering. *In* GI/ITG Workshop MMBnet, pages 13-14. Cité 1 fois, p. 34.

Harrou, F., Kadri, F., Chaabane, S., Tahon, C. et Sun, Y. (2015). Improved principal component analysis for anomaly detection: Application to an emergency department. *In* Computers & Industrial Engineering, Vol. 88, pages 63–77. Elsevier. Cité 1 fois, p. 35.

Ringberg, H., Soule, A., Rexford, J. et Diot, C. (2007). Sensitivity of PCA for traffic anomaly detection. *In Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 109-120. Cité 1 fois, p. 35.

Bereziński, p., Jasiul, B. et Szpyrka, M. (2015). An entropy-based network anomaly detection method. Entropy, Vol. 17(4), pages 2367-2408. Cité 1 fois, p. 35.

Nychis, G., Sekar, V., Andersen, D.G., Kim, H. et Zhang, H. (2008). An empirical evaluation of entropy-based traffic anomaly detection. *In Proceedings of the ACM SIGCOMM conference on Internet measurement*, pages 151-156. Cité 1 fois, p. 35.

Han, T., Lan, Y., Xiao, L, Huang, B. et Zhang, K. (2014). Event detection with vector similarity based on fourier transformation. *In ICCSSE'14: Proceedings of IEEE International Conference on Control Science and Systems Engineering*, pages 195-199. IEEE. Cité 1 fois, p. 35.

Du, Z., Ma, L., Li, H., Li, Q, Sun, G. et Liu, Z. (2018). Network traffic anomaly detection based on wavelet analysis. *In SERA'18: Proceedings of IEEE 16th International Conference on Software Engineering Research, Management and Applications*, pages 94–101. IEEE. Cité 1 fois, p. 35.

Lu, M. et Ghorbani, M. (2008). Network anomaly detection based on wavelet analysis. *In* EURASIP Journal on Advances in Signal Processing, pages 1-16. Springer. Cité 1 fois, p. 35.

Hardin, J. et Rocke, D.M. (2004). Outlier detection in the multiple cluster Setting using the minimum covariance determinant estimator. *In* Computational Statistics and Data Analysis, vol. 44(4), pages 625-638.Elsevier. Cité 1 fois, p. 34.

Xu, X., Liu, H. et Yao, M. (2019). Recent progress of anomaly detection. Complexity. Hindawi. Cité 1 fois, p. 27.

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G. et Vázquez, E. (2009). Anomaly based network intrusion detection: Techniques, systems and challenges. computers & security, vol. 28(1-2), pages 18–28, Elsevier. Cité 1 fois, p. 28.

Zaki, M.J. et Meira, M. (2014). Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press Cité 2 fois, p. 28 et 33.

Däubener, S., Schmitt, S. et Wang, H. et Bäck, T. (2019). Anomaly Detection in Univariate Time Series: An Empirical Comparison of Machine Learning Algorithms. *In ICDM'19: 19th Industrial Conference on Data Mining ICDM 2019.* Cité 2 fois, p. 20 et 27.

Braei, M. et Wagner, S. (2020). Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art. arXiv preprint arXiv:2004.00433. Cité 1 fois, p. 27.

[Blázquez-García, A., [Conde, A., [Mori, U. et Lozano, J.A. (2020). A review on outlier/anomaly detection in time series data. arXiv preprint arXiv:2002.04236. Cité 2 fois, p. 19 et 27.

Feremans, L., Vercruyssen, V., Cule, B., Meert, W. et Goethals, B. (2019). Pattern-based anomaly detection in mixed-type time series. *In Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 240-256. Springer, Cham. Cité 4 fois, p. 25, 33, 38 et 107.

Abe, K. et Hori, H. (2003). An effective pattern based outlier detection approach for mixed attribute data. *In Australasian joint conference on artificial intelligence*, pages 122-131. Springer, Berlin, Heidelberg. Cité 1 fois, p. 33.

He, Z., Xu, X., Huang, Z.J. et Deng, S. (2005). FP-outlier: Frequent pattern based outlier detection. Computer Science and Information Systems, vol. 2(1), pages 103-118. Cité 1 fois, p. 33.

Ghoting, A., Otey, M.E. et Parthasarathy, S. (2004). LOADED: Link-based outlier andanomaly detection in evolving data sets. *In ICDM'04: Fourth IEEE International Conference on Data Mining*, pages 387-390. IEEE. Cité 1 fois, p. 33.

yun Chen, X.Y. et yan Zhan, Y.Y. (2008). Multi-scale anomaly detection algorithm based on infrequent pattern of time series. Journal of Computational and Applied Mathematics, vol. 214(1), pages 227-237. Cité 4 fois, p. 25, 33, 38 et 107.

Kuchar, J. et Svátek, V. (2018). Spotlighting anomalies using frequent patterns. *In SAINT'03: KDD 2017 Workshop on Anomaly Detection in Finance*, pages 33-42. Cité 1 fois, p. 33.

Abghari, S., Boeva, V., Lavesson, N., Grahn, H., Gustafsson, J. et Shaikh, J. (2018). Outlier Detection for Video Session Data Using Sequential Pattern Mining. *In ACM SIGKDD Workshop On Outlier Detection De-constructed*, London. Cité 1 fois, p. 33.

Bouasker, S. et Ben Yahia, S. (2015). Key correlation mining by simultaneous monotone and anti-monotone constraints checking. *In SAC'15: Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 851-856, New York, NY, USA. ACM Press. Cité 1 fois, p. 33.

Rahman, A., Xu, Y., Radke, K. et Foo, E. (2016). Finding anomalies in SCADA logs using rare sequential pattern mining. *In International Conference on Network and System Security*, pages 499-506. Springer, Cham. Cité 1 fois, p. 33.

Estevez-Tapiador, J.M., Garcia-Teodoro, P. et Diaz-Verdejo, J.E. (2004). Anomaly detection methods in wired networks: a survey and taxonomy. Computer Communications, vol. 27(16), pages 1569-1584. Springer, Cham. Cité 1 fois, p. 28.

Sekar, R., Gupta, A., Frullo, J., Shanbhag, T., Tiwari, A., Yang, H. et Zhou, S. (2002). Specification-based anomaly detection: a new approach for detecting network intrusions. *In CCS'02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 265-274, New York, NY, USA. ACM Press. Cité 1 fois, p. 28.

Fox, A.J (1972). Outliers in Time Series. Journal of the Royal Statistical Society: Series B (Methodological), vol. 34(3), pages 350–363. Cité 1 fois, p. 19.

Tsay, R.S..... (1988). Outliers, Level Shifts, and Variance Changes in Time Series. Journal of Forecasting, vol. 7(1), pages 1–20. Cité 3 fois, p. 19, 89 et 90.

Tsay, R.S., Peña, D. et Pankratz, A.E. (2000). Outliers in multivariate time series. *Biometrika*, vol. 87(4), pages 789–804. Cité 1 fois, p. 19.

Carrera, D., Rossi, B., Fragneto, P. et Boracchi, G. (2019). Online anomaly detection for long-term ECG monitoring using wearable devices. *Pattern Recognition*, vol. 88, pages 482-492. Cité 1 fois, p. 19.

Mehrotra, K.G., Mohan, C.K. et Huang, H. (2017). Anomaly Detection Principles and Algorithms. *Terrorism, Security, and Computation*, page 217. New York, NY, USA:Springer International Publishing. Cité 1 fois, p. 21.

Wu, H.S. (2016). A survey of research on anomaly detection for time series. *In ICCWAMTIP'16: 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing*, pageS 426-431. IEEE Computer Society. Cité 1 fois, p. 27.

Sokolova, M., Japkowicz, N. et Szpakowicz, S. (2006). Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. *In Australasian joint conference on artificial intelligence*, pages 1015-1021. Springer, Berlin, Heidelberg. Cité 1 fois, p. 36.

Tatbul, N., Lee, T.J., Zdonik, S., Alam, M. et Gottschlich, J. (2018). Precision and recall for time series. *In Advances in Neural Information Processing Systems*, pages 1920-1930. Cité 1 fois, p. 37.

Yeh, C.C.M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H.A., ... et Keogh, E. (2016). Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. *In ICDM'16: 2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317-1322. IEEE. Cité 4 fois, p. 25, 33, 38 et 107.

Agrawal, S. et Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, vol. 60, pages 708-713. Cité 1 fois, p. 27.

Omar, S., Ngadi, A. et Jebur, H.H. (2013). Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications*, vol. 79(2). Cité 1 fois, p. 28.

Lakshmi, K.N., Neema, N., Mohammed Muddasir, N. et Prashanth, M.V. (2020). Anomaly Detection Techniques in Data Mining—A Review. *In Inventive Communication and Computational Technologies*, vol. 89, pages 799-804. Springer, Singapore. Cité 1 fois, p. 28.

Barakat, N. et Diederich, J. (2005). Eclectic rule-extraction from support vector machines. *International Journal of Computational Intelligence*, vol. 2(1), pages 59-62. Citeseer. Cité 2 fois, p. 64 et 78.

Singh, S. et Gupta, P. (2014). Comparative study ID3, cart and C4. 5 decision tree algorithm: a survey. *IJAIST'14: International Journal of Advanced Information Science and Technology*, vol. 27(27), pages 97-103. Cité 2 fois, p. 64 et 73.

Munir, M., Erkel, S., Dengel, A. et Ahmed, S. (2017). Pattern-based contextual anomaly detection in hvac systems. *In ICDMW'17: 2017 IEEE International Conference on Data Mining Workshops*, pages 1066-1073. IEEE. Cité 1 fois, p. 50.

Su, J. et Zhang, H. (2006). A fast decision tree learning algorithm. *AAAI*, vol. 6, pages 500-505. Cité 1 fois, p. 72.

Daud, N.R. et Corne, W. (2009). Human readable rule induction in medical data mining. *In Proceedings of the European Computing Conference*, vol. 27(1), pages 787-798. Springer, Boston, MA. Cité 1 fois, p. 78.

Snoek, J., Larochelle, H. et Adams, R.P. (2012). Practical bayesian optimization of machine learning algorithms. *In Advances in neural information processing systems*, vol. 17(1), pages 2951-2959. Cité 2 fois, p. 80 et 81.

Wu, J., Chen, X.Y., Zhang, H., Xiong, L.D., Lei, H. et Deng, S.H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *In Journal of Electronic Science and Technology*, pages 26-40. Elsevier. Cité 1 fois, p. 80.

Xia, Y., Liu, C., Li, Y. et Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *In Expert Systems with Applications*, VOL. 78, pages 225-241. Elsevier. Cité 1 fois, p. 80.

Su, N. et Zhang, S. (2015). A labeled anomaly detection dataset s5 yahoo research. *Available: https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70*, v1.[Online]. Cité 2 fois, p. 90 et 104.

Ramanathan, N., Balzano, L., Burt, M. et Estrin, D. et al.(2006). Rapid deployment with confidence: Calibration and fault detection in environmental sensor networks. *UCLA: Center for Embedded Network Sensing.* Cité 1 fois, p. 94.

Hochenbaum, J., Vallis, O. S., Kejariwal, A. (2017) Automatic anomaly detection in the cloud via statistical learning. *arXiv preprint arXiv:1704.07706.* Cité 3 fois, p. 29, 38 et 95.

Cleveland, R. B., Cleveland, W. S., McRae, J. E. et Terpenning, I. (1990). STL: A seasonal-trend decomposition. *In Journal of official statistics*, VOL. 6(1), pages 3-73. Cité 1 fois, p. 94.

López-de-Lacalle, J. (2016). tsoutliers R package for detection of outliers in time series. *CRAN, R Package.* Cité 1 fois, p. 95.

Frank, E. et Witten, I.H. (1998). Generating accurate rule sets without global optimization. *In 15th ICML*, pages 144-151. University of Waikato, Department of Computer Science. Cité 1 fois, p. 109.

Frank, W. W. (1995). Fast effective rule induction. *In the 12th International Conference on Machine Learning*, pages 115–123. Elsevier. Cité 1 fois, p. 109.

Witten, I.H. et Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques–2nd ed. Morgan Kaufmann series in data management systems. Cité 1 fois, p. 110.

Taylor, G. George Boole 1815-1864. *In In Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences*, vol. 57 pages. 66-73, Royal Irish Academy. Cité 1 fois, p. 77.

Agrawal, R. et Srikant, R. (1994). Fast algorithms for mining association rules. *In Proc. 20th int. conf. very large data bases, VLDB*, vol. 1215, pages. 487-499. Cité 1 fois, p. 14.

Ahmed, M., Mahmood, A. N. et Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. *In Future Generation Computer Systems*, VOL. 55, pages 278-288. Elsevier. Cité 1 fois, p. 22.

Akoglu, L. et Faloutsos, C. (2013). Anomaly, event, and fraud detection in large network datasets. *In Proceedings of the sixth ACM international conference on Web search and data mining*, pages. 773-774. Cité 1 fois, p. 22.

Ukil, A., Bandyoapdhyay, S., Puri, C. et Pal, A. (2016). IoT healthcare analytics: The importance of anomaly detection. *In 2016 IEEE 30th international conference on advanced information networking and applications (AINA)*, pages 994-997. IEEE. Cité 1 fois, p. 22.

Hauskrecht, M., Valko, M., Kveton, B., Visweswaran, S. et Cooper, G. F. (2007). Evidence-based anomaly detection in clinical domains. *In AMIA Annual Symposium Proceedings*, pages 319. American Medical Informatics Association. Cité 1 fois, p. 22.

Purarjomandlangrudi, A., Ghapanchi, A. H. et Esmalifalak, M. (2014). A data mining approach for fault diagnosis: An application of anomaly detection algorithm. *Measurement*, vol. 55, pages 343-352. Elsevier. Cité 1 fois, p. 22.

Rajasegarar, S., Leckie, C. et Palaniswami, M. (2008). Anomaly detection in wireless sensor networks. *IEEE Wireless Communicationst*, vol. 15(4), pages 34-40. IEEE. Cité 1 fois, p. 22.

Hayes, M. A. et Capretz, M. A. (2014). Contextual anomaly detection in big sensor data. *In 2014 IEEE International Congress on Big Data*, pages. 64-71. IEEE. Cité 1 fois, p. 22.

Rabatel, J., Bringay, S. et Poncelet, P. (2011). Anomaly detection in monitoring sensor data for preventive maintenance. *Expert Systems with Applications*, vol. 38(6), pages 7003-7015. Elsevier. Cité 1 fois, p. 22.

Au, C. E., Skaff, S. et Clark, J. J. (2006). Anomaly detection for video surveillance applications. *In 18th International Conference on Pattern Recognition (ICPR'06)*, vol. 4, pages 888-891. IEEE. Cité 1 fois, p. 22.

Sabokrou, M., Fayyaz, M., Fathy, M., Moayed, Z. et Klette, R. (2018). Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, vol. 172, pages 88-97. Elsevier. Cité 1 fois, p. 22.

Othman, Z. et Eshames, H. F. (2012). Abnormal patterns detection in control charts using classification techniques. *In J Adv Comput Technol*, vol. 4(10), pages. 61-70. IEEE. Cité 1 fois, p. 22.

Kim, T. Y. et Cho, S. B. (2018). Web traffic anomaly detection using C-LSTM neural networks. *In Expert Systems with Applications*, vol. 106, pages. 66-76. Elsevier. Cité 1 fois, p. 31.

El Malki, N., Ravat, F. et Teste, O. (2020). KD-means: clustering method for massive data based on kd-tree. *In Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP)*, vol. 2572, pages 26-35. Elsevier. Cité 1 fois, p. 35.

Cité 1 fois, p. 35.
Mallat, S. (2000). Une exploration des signaux en ondelettes. Editions Ecole Polytechnique.

Ben Kraiem, I., Ghozzi , F., Peninou, A. et Teste, O. (2019). CoRP: A Pattern-Based Anomaly Detection in Time-Series. *In International Conference on Enterprise Information Systems (ICEIS)*, vol. 378, pages 424-442. Springer. Cité 1 fois, p. 60.

Ben Kraiem, I., Ghozzi , F., Peninou, A. et Teste, O. (2019). Pattern-based method for anomaly detection in sensor networks. *In Proceedings of the 21st International Conference on Enterprise Information Systems*, vol. 1, pages 104–113. SciTePress. Cité 2 fois, p. 38 et 60.

Ben Kraiem, I., Ghozzi , F., Peninou, A. et Teste, O. (2019). Méthode à base de patterns pour la détection d'anomalies. *In 37e Congres Informatique des Organisations et Systemes d'Information et de Decision (INFORSID 2019)*, pages 239-254. Cité 1 fois, p. 60.

Ben Kraiem, I., Ghozzi , F., Peninou, A., Roman-Jimenez, G. et Teste, O. (2020). Automatic Classification Rules for Anomaly Detection in Time-Series. *In International Conference on Research Challenges in Information Science (RCIS)*, pages 321-337. Springer. Cité 1 fois, p. 83.

Breiman, L., Friedman, J., Olshen, R.A. et Stone, C. J. (1984). Classification and Regression Trees. *Wadsworth and Brooks*. CRC press. Cité 1 fois, p. 72.

Breiman, J.R. (1993). C4.5: programs for machine learning. . Elsevier. Cité 1 fois, p. 73.

Ray, A. (2018). Compassionate Artificial Intelligence: Frameworks and Algorithms. . Compassionate AI Lab (An Imprint of Inner Light Publishers). Cité 1 fois, p. 1.

Petkovic, M., Mihajlovic, V., Jonker, W. et Djordjevic-Kajan, S. (2002). Multi-modal extraction of highlights from TV formula 1 programs. *In Proc. IEEE ICME*, pages 817–820. Cité 1 fois, p. 31.

Hota, H. S., Handa, R., Shrivas, A.K. (2017). Time series data prediction using sliding window based rbf neural network. *International Journal of Computational Intelligence Research*, vol. 13(5), pages 1145-1156. Cité 1 fois, p. 14.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, vol. 104(1), pages 148-175. Cité 1 fois, p. 81.

Ben Kraiem, I., Ghozzi , F., Peninou, A., Roman-Jimenez, G. et Teste, O. (2021). Human-Interpretable Rules for Anomaly Detection in Time-series. *In Proceedings of the 24th International Conference on Extending Database Technology, EDBT 2021.* Cité 1 fois, p. 83.

Gançarski , P., Lafabregue, B., Salaou, A.D. et Vernier, H. (2020). FODOMUST-Une plateforme de clustering collaboratif sous contraintes incrémental de séries temporelles. *In EGC*, pages 507-514. Cité 1 fois, p. 34.

Owuor , D. O., Laurent, A. et Orero, J. O. (2020). Exploiting IoT data crossings for gradual pattern mining through parallel processing. *In ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium*, pages 110-121. Springer, Cham. Cité 1 fois, p. 33.

Zuo , J., Zeitouni, K. et Taher, Y. (2018). SE2TeC: A Scalable Engine for Efficient and Expressive Time Series Classification. *In BDCSIntell*, pages 8-11. Cité 1 fois, p. 25.

Liste des figures

Liste des tableaux